FABRICATION AND PROTOTYPING OF UNMANNED GROUND VEHICLE FOR OFF-ROAD APPLICATIONS USING DEEP LEARNING

A PROJECT REPORT

Submitted by

BLESSED SACHIN EMMANUEL PAUL RAJ (2019502007)

KADHIR T (2019502017)

KEERTHIVASAN S (2019502019)

MOHAMMED MARZOOK ABDULLAH (2019502029)

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

AUTOMOBILE ENGINEERING

MADRAS INSTITUTE OF TECHNOLOGY, CHROMPET, CHENNAI-44

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2023

BONAFIDE CERTIFICATE

Certified that this project report **"FABRICATION AND PROTOTYPING OF UNMANNED GROUND VEHICLE FOR OFF ROAD APPLICATION USING DEEP LEARNING"** is the Bonafide work of **"BLESSED SACHIN EMMANUEL PAUL RAJ, KADHIR T, KEERTHIVASAN S, MOHAMMED MARZOOK ABDULLAH"** who carried out the project work under my supervision.

SIGNATURE

Dr. K. ANNAMALAI

HEAD OF DEPARTMENT
Automobile Engineering
Madras Institute of Technology
Anna University
Chennai-600 044

SIGNATURE

Dr. S. ARUL SELVAN

SUPERVISOR
Assistant Professor (Sl. Gr.)
Automobile Engineering
Madras Institute of Technology
Anna University
Chennai-600 044

ABSTRACT

Autonomous vehicles have long been in the realm of science fiction, however recent progress means that these driverless cars will be on our streets in the relatively near future. There is strong competition between newer technology companies (such as Google, Uber and Tesla) and established car companies (such as Mercedes Benz, General Motors, Nissan and many others). Some have been working on autonomous vehicles for years, and there are many working prototypes and trial programs. Obviously, there are still aspects of the driverless car that still need to be refined, and there are many legal, liability, technical and social problems that must be overcome. However, in terms of transport planning into the future, autonomous vehicles should be considered, as they are likely to have significant impacts on travel behavior and road network operations. This paper will address current progress and direction for autonomous vehicles, what this could mean for the future of transport and the possible analytical approaches to addressing these impacts. Ever since vehicles were first invented, futurists have been thinking about taking humans out of the driver's seat. Between 1920 and 1980 many efforts had been made by various car companies and Universities to pioneer autonomous vehicles. One of the first demonstrations was a radio-controlled driver-less car in the 1920's. This still required a second car behind to send out radio signals to the transmitting antenna that was installed in the 'driverless' vehicle in front (The Milwaukee Sentinel 1926). A few decades later, people considered driverless cars that could be activated by electronic devices embedded in the roadway. This would mean construction of new electronically controlled streets; these were considered in the UK and parts of the US. After early enthusiasm, the funding was withdrawn in both cases. Since the early 2000's, many universities and car companies have been working on improving vehicle autonomy. Although they worked most of the time, sometimes a human driver had to intervene and navigating intersections was difficult (see the next section for discussion on the levels of autonomy). Google is one of many companies that have had success with autonomous vehicles. Improvements are still being made today to get vehicles to operate fully autonomously, whilst making sure safety is maintained, and improved where possible.

Table of Contents

Table of Figures

Table of Tables

CHAPTER 1

1.1 INTRODUCTION

An autonomous robot is a robot that acts without recourse to human control. The first autonomous robot environments were known as Elmer and Elsie, which were constructed in the late 1940s by W. Grey Walter. They were the first robots in history that were programmed to "think" the way biological brains do and meant to have free will. Elmer and Elsie were often labelled as tortoises because of how they were shaped and the manner in which they moved. They were capable of phototaxis which is the movement that occurs in response to light stimulus.[citation needed]
Historic examples include space probes. Modern examples include self-driving vacuums and cars. Industrial robot arms that work on assembly lines inside factories may also be considered autonomous robots, though their autonomy is restricted due to a highly structured environment and their inability to locomote.

Self-maintenance

The first requirement for complete physical autonomy is the ability for a robot to take care of itself. Many of the battery-powered robots on the market today can find and connect to a charging station, and some toys like Sony's Aibo are capable of self-docking to charge their batteries.
Self-maintenance is based on "proprioception", or sensing one's own internal status. In the battery charging example, the robot can tell proprioceptively that its batteries are low and it then seeks the charger. Another common proprioceptive sensor is for heat monitoring. Increased proprioception will be required for robots to work autonomously near people and in harsh environments. Common proprioceptive sensors include thermal, optical, and haptic sensing, as well as the Hall effect (electric).

Sensing the environment

Exteroception is sensing things about the environment. Autonomous robots must have a range of environmental sensors to perform their task and stay out of trouble. The autonomous robot can recognize sensor failures and minimize the impact on the performance caused by failures.
Common exteroceptive sensors include the electromagnetic spectrum, sound, touch, chemical (smell, odor), temperature, range to various objects, and altitude.

Some robotic lawn mowers will adapt their programming by detecting the speed in which grass grows as needed to maintain a perfectly cut lawn, and some vacuum cleaning robots have dirt detectors that sense how much dirt is being picked up and use this information to tell them to stay in one area longer.

Task performance

The next step in autonomous behaviour is to actually perform a physical task. A new area showing commercial promise is domestic robots, with a flood of small vacuuming robots beginning with iRobot and Electrolux in 2002. While the level of intelligence is not high in these systems, they navigate over wide areas and pilot in tight situations around homes using contact and non-contact sensors. Both of these robots use proprietary algorithms to increase coverage over simple random bounces.
The next level of autonomous task performance requires a robot to perform conditional tasks. For instance, security robots can be programmed to detect intruders and respond in a particular way depending upon where the intruder is. For example, Amazon (company) launched its Astro for home monitoring, security and eldercare in September 2021.
Autonomous navigation

Indoor navigation

For a robot to associate behaviours with a place (localization) requires it to know where it is and to be able to navigate point-to-point. Such navigation began with wire-guidance in the 1970s and progressed in the early 2000s to beacon-based triangulation. Current commercial robots autonomously navigate based on sensing natural features. The first commercial robots to achieve this were Pyxus' HelpMate hospital robot and the Cyber Motion guard robot, both designed by robotics pioneers in the 1980s. These robots originally used manually created CAD floor plans, sonar sensing and wall-following variations to navigate buildings. The next generation, such as Mobile Robots' Patrol Bot and autonomous wheelchair,[4] both introduced in 2004, have the ability to create their own laser-based maps of a building and to navigate open areas as well as corridors. Their control system changes its path on the fly if something blocks the way.
At first, autonomous navigation was based on planar sensors, such as laser range-finders, that can only sense at one level. The most advanced systems now fuse information from various sensors for both localization (position) and navigation. Systems such as Motivity can rely on different sensors in different areas, depending

upon which provides the most reliable data at the time, and can re-map a building autonomously.

Rather than climb stairs, which requires highly specialized hardware, most indoor robots navigate handicapped-accessible areas, controlling elevators, and electronic doors.[5] With such electronic access-control interfaces, robots can now freely navigate indoors. Autonomously climbing stairs and opening doors manually are topics of research at the current time.
As these indoor techniques continue to develop, vacuuming robots will gain the ability to clean a specific user-specified room or a whole floor. Security robots will be able to cooperatively surround intruders and cut off exits. These advances also bring concomitant protections: robots' internal maps typically permit "forbidden areas" to be defined to prevent robots from autonomously entering certain regions.

Outdoor navigation

Outdoor autonomy is most easily achieved in the air, since obstacles are rare. Cruise missiles are rather dangerous highly autonomous robots. Pilotless drone aircraft are increasingly used for reconnaissance. Some of these unmanned aerial vehicles (UAVs) are capable of flying their entire mission without any human interaction at all except possibly for the landing where a person intervenes using radio remote control. Some drones are capable of safe, automatic landings, however. SpaceX operates a number of Autonomous spaceport drone ships, used to safely land and recover Falcon 9 rockets at sea.[6]
Outdoor autonomy is the most difficult for ground vehicles, due to:
Three-dimensional terrain
Great disparities in surface density
Weather exigencies
Instability of the sensed environment

1.2.CONVOLUTIONAL NEURAL NETWORK

In deep learning, a convolutional neural network is a class of artificial neural network (ANN), most applied to analyse visual imagery. CNNs are also known as Shift Invariant or Space Invariant Artificial Neural Networks (SIANN), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation-equivariant responses known as feature maps. Counter-intuitively, most convolutional neural networks are not invariant to translation, due to the down sampling operation they apply to the input. They have applications in image and video recognition, recommender systems, image classification, image segmentation, medical image analysis, natural language processing, brain–computer interfaces, and financial time series.[8]

CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks make them prone to overfitting data. Typical ways of regularization, or preventing overfitting, include: penalizing parameters during training (such as weight decay) or trimming connectivity (skipped connections, dropout, etc.) CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble patterns of increasing complexity using smaller and simpler patterns embossed in their filters. Therefore, on a scale of connectivity and complexity, CNNs are on the lower extreme.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

1.3. INSPIRATION FOR OUR WORK

Goliath was the codename for a small, remote-controlled tank developed by the Nazi Germany during World War II. It was designed for use in urban warfare and was intended to provide infantry with mobile firepower and reconnaissance capabilities.

The Goliath was powered by an electric motor and was steered by cables that ran from a control box to the vehicle. It was armed with either a high-explosive charge or an anti-tank mine and could carry up to 100 kilograms of explosives.

Despite its potential, the Goliath had several significant drawbacks. It was expensive to produce and was prone to malfunctioning, and its small size made it vulnerable to damage from small arms fire. Additionally, its control system required a clear line of sight between the operator and the vehicle, which limited its effectiveness in dense urban environments.

Overall, while the Goliath represented an innovative use of remote-controlled technology, it ultimately had limited impact on the outcome of the war.

It's interesting to hear that the Goliath has inspired our project on autonomous unmanned ground vehicles. While the Goliath had its limitations, it was an innovative use of remote-controlled technology at the time and served as a predecessor to modern unmanned ground vehicles.

Today's autonomous unmanned ground vehicles are equipped with advanced sensing, computing, and communication technologies, allowing them to operate with greater autonomy and precision. They have a wide range of applications, including military and civilian uses such as surveillance, logistics, search and rescue, and agriculture.

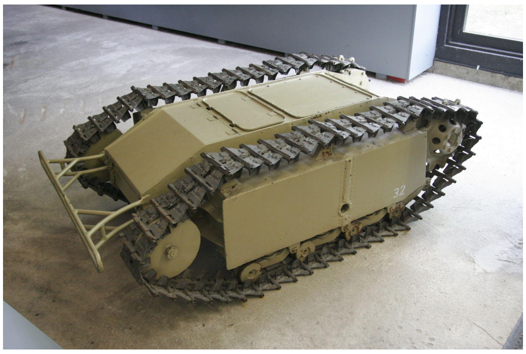

Figure 1.1 Goliath developed by Nazi

CHAPTER 2

2.1 LITERATURE SURVEY

A literature review gave a comprehensive summary of previous research on this topic. The literature review surveys scholarly articles, books, and other sources relevant to this particular area of research. The review will enumerate, describe, summarize, objectively evaluate and clarify this previous research. It will give a theoretical base for the research and help the project determine the nature of research. The literature review acknowledges the work of previous researchers, and in so doing, assures the reader that the work has been well conceived.

1. **Barrozo ,V. Lazcano,5th International Conference on Frontiers of Signal Processing (2019)**
   Processed images are inputs for a control strategy that defines commands to the Remote Terrestrial Vehicle (RTV). Our proposal was tested using CGI images. Those CGI images represent a road with inserted obstacles.

2. **Farid Bounini, Denis Gingras, Vincent Lapointe, Herve Pollart (2019)**
   A Canny edges' detector is used to determine frames' edges, and essentially the road boundaries and painted lines.

3. **Krämer, Marc StevenKuhnert, Lars Kuhnert, Klaus-Dieter,26th International Conference in Central Europe on Computer Graphics(2018)**
   In urban regions, roads can be detected by lane markers or delimitations. Developed a software system that is based on mostly simple and low computationally intensive algorithms. We developed and tested the functions with a large dataset of camera images and also generated a manually Ground Truth for the evaluation.

4. **Hailong Qin, Zehui Meng, Wei Meng, Xudong Chen, Hao Sun, Feng Lin, Marcelo H. Ang Jr IEEE Transactions on Vehicular Technology (Volume: 68, Issue: 2, February 2019)**
   One of the most demanding capabilities of the autonomous vehicles is being able to navigate and execute tasks autonomously in complex environments, especially in GPS-denied environments.

5. **Research Numerical of the high-speed military vehicle track Zdzisław Hryciów1, a) and Piotr Rybak1,Military University of Technology, Faculty of Mechanical Engineering, Institute of Motor Vehicles and Transportation. Published Online: 04 March 2019**
   The forces in track links reach significant values. Therefore, an attempt was made to determine the value of dynamic forces acting on tank during overcoming selected terrain obstacles. For this purpose, a numerical model of the T-72 tank was developed. With its use, simulation studies were carried out, including overcoming typical obstacles.

6. **Optimization of autonomous vehicle speed control mechanisms using hybrid DDPG-SHAP-DRL-stochastic algorithm (by syasavya ,Volume 173, November 2022, 103245)**
   Autonomous Vehicles (AV) are the future milestones of the automobile industry, which functions without the intervention of human being. Numerous researches have been stimulated by leading automobile sectors of the world, to address the anticipated challenges in implementing the autonomous vehicles in a practical scenario. The speed control mechanism is the predominant challenge which acts in the basis of Machine Learning mechanism is the major thrust area associated with autonomous vehicles. Reinforcement Learning (RL) is the effective algorithm to solve the challenges associated with the autonomous driving of vehicles and its decision on complex scenarios. A simulative environment is advantageous for training and validation of an RL algorithm because it reduces risk and saves resources. This research work introduces a novel hybrid algorithm composed of Deep Deterministic Policy Gradient (DDPG) – Shapley Additive explanations (SHAP) – Deep Reinforcement Learning (DRL)-stochastic algorithm. The primary objective of this research work is to introduce an RL environment for optimizing longitudinal control

7. **Driving Behaviour of L3 Autonomous Vehicle Drivers in Fog Zones under Different Traffic Flow Conditions (by Haijian Li Yuxuan Li Kaiqun Chen Xiaohu Zhaoa, 7 December 2022, 112300)**
   The foggy freeway is a scenario with a high incidence of accidents. Different traffic flow conditions will affect the driving behaviour of conditionally autonomous vehicle drivers in the foggy areas of the freeway. In this study, the experiment was carried out by using the driving simulation system. Forty-two participants drove on a foggy freeway according to two traffic conditions and two non-driving related tasks. Wilcoxon Signed-Rank Test and Survival Analysis were applied to explore the impact. The results show that the effects of different non-driving related tasks on driving behaviour are mainly concentrated in the takeover process; The deceleration reaction time of the free flow (6.04s) is much lower than that of bound flow (9.40s); In the condition of bound flow, the driver is more inclined to slow down without braking. Potential

application areas of this research comprise safety assessment of conditional autonomous driving and formulating autonomous driving policy.

8. **Learning dynamic background for weakly supervised moving object detection Author links open overlay (by ZhijunZhang, , May 2022, 104425)** Moving Object Detection (MOD) aims at extracting foreground moving objects in videos from static cameras. While low-rank based approaches have achieved impressive success in the MOD task, their performance remains limited on dynamics background scenes. The main reason is that dynamic clutters, *e.g.*, *swaying leaves and rippers*, are easy to mix up with moving objects in the decomposition model which simply classify the sparse noise as foregrounds. In order to improve the generalization ability of low-rank based moving object detectors, we suggest adding an explicit dynamic clutter component in the decomposition framework with realistic dynamic background modeling. Then the dynamic clutter can be learned through object-free video data due to their self-similarity across time and space. Thus, the moving objects can be naturally separated by a tensor-based decomposition model which formulates the static background by a unidirectional low-rank tensor, learns the dynamic clutter by a two-stream neural network, and constrains moving objects with spatiotemporal continuity. To further provide a more accurate object detection result, an object prior is embedded into our model in an attention manner. Extensive experimental results on the challenging datasets of dynamic background clearly demonstrate the superior performance of our model over the state-of-the-art in terms of quantitative metrics and visual quality.

9. **Measurement of the aerodynamic and rolling resistances of road tanker vehicles from coast-down testsEM Evans, PJ Zemroch, *Proceedings of the Institution of Mechanical Engineers, Part D: Transport Engineering 198 (3), 211-218, 2014*** The aerodynamic and rolling resistances of three articulated road tankers have been measured in a statistically planned series of coast-down tests. The aerodynamic drag coefficient, CD, was found to be similar for all three vehicles and CD, did not depend on their loading conditions. The rolling resistance, R, approximately doubled when the vehicle was loaded to three times its unladen weight
10. **Magnesium and aluminum alloys in automotive industry ,AH Musfirah, AG Jaharah ,Journal o Applied Sciences Research 8 (9), 4865-4875, 2018** Today's interest is focusing on growing demand for more fuel-efficient vehicles to reduce energy consumption and air pollution which become a challenge for the current automotive industry. Driven by this requirement, researchers had done many researches to find a suitable material that fulfil the requirement for automotive parts. In this paper, general application of magnesium and aluminum alloys in automotive industry is presented, especially about the material properties, machinability and cost comparison of these alloys. In

addition, the current and potential automotive applications of aluminium are reviewed, and the technical challenges for these applications are also discussed. Based on the previous studies, it was found that the aluminium has several advantages over magnesium in terms of manufacturability due to its mechanical and physical properties.

11. **Rubber Pads for Tank Track ,Edward W Bergstrom, John R Cerny - ARMY WEAPONS COMMAND ROCK ISLAND IL RESEARCH DEVELOPMENT AND ENGINEERING DIRECTORATE, 2010**
    Investigations of rubber compounding, service test evaluation, and rubber-to-metal bonding were made. The injection moulding of track pads was attempted. Numerous compounds were developed for optimum properties, and track pads prepared from these compounds were service tested to determine actual wear resistance. Long-term aging tests on mil-able polyester urethane track pads and rubber-to-metal bonded specimens were completed. Compounds based on Stereon 750, HYTRANS elastomers, SBR polybutadiene blends, and EPDM provided pads with improved tread wear. Little correlation was found between volume wear ratings based on service tests for cut crack growth, heat build-up, tear resistance, and abrasion resistance. Track pads prepared by injection moulding had physical properties comparable to those of compression-moulded pads. The preparation of rubber track pads, having significantly improved wear resistance, from certain low-cost, general-purpose type elastomers appears feasible.

12. **Detect the mechanical structure of the vehicle with neural network, Hui-Hung Chien Department of Electrical Engineering, National Sun Yat-sen University Kaohsiung, Taiwan - hhchien@g-mail.nsysu.edu.tw**
    With the rise of consumers ’awareness of driving safety and the popularity of automated vehicles, people pay highly attention to vehicle data. Vehicle data is mainly divided into the following two types: “vehicle internal information” (engine, electrical equips, transmission components) , “the communication between the vehicle and the external environment” (vehicle driving safety, driving experience). These type of data can be collected through the vehicle's on-board diagnostics (OBD) system. The data set will be used for analysis. The vehicle's driving profile in different states provides a series of vehicle safety services.

CHAPTER 3

3.1 PROJECT TITLE

“ FABRICATION AND PROTOTYPING OF UNMANNED GROUND VEHICLE FOR OFF-ROAD APPLICATIONS USING DEEP LEARNING”

3.2 AIM OF THE PROJECT

To develop a working scale model of an unmanned ground vehicle Our project focuses on automation of such vehicles using Raspberry-Pi through computer vision and deep learning techniques to detect off road lanes and obstacles and classify them as either immoral or moral to climb over. An autonomous vehicle, or a driverless vehicle, is one that is able to operate itself and perform necessary functions without any human intervention, through the ability to sense its surroundings.

3.3 OBJECTIVES

- To Design and develop algorithm for off-road detection and navigation .
- Detect , classify and approximate the dimension of the objects and to be able to distinguish
  as ethical or unethical to climb over .
- To Fabricate a prototype , test and optimise the prototype to achieve higher level of automation

3.4 OBJECT DETECTION AND CLASSIFICATION

Object detection combines classification and localization to determine what objects are in the image or video and specify where they are in the image. It applies classification to distinct objects and uses bounding boxes Our Prototype uses fast-R CNN based object detection algorithm to identify and locate various objects or obstacles and take decision either to move over or avoid the obstacle based on the safety parameters for example a rock can be climbed over but a sharp nail or any small animal must be avoided.

3.5 DIMENSION DETECTION

The dimensions of an obstacle are measured using a monocular camera with help of a reference entity mounted on the chassis and using the pixel counting technique to assert in decisions to drive over an object based on ground clearance of the vehicle.

3.6 PATH DETECTION

Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. The pixel inside the road-like feature is selected as a reference value using which pixel Grouping algorithm is implemented to segment different patches in an image .

The top five countries promoting autonomous vehicles are global powerhouses, namely USA, Japan,France,United Kingdom and Germany. However, USA is AV (Autonomous Vehicle) ready by a great margin compared to the other countries.

In the Indian context, self-driving cars are both a tantalising prospect and a dystopian nightmare. On one hand it can change the way we live, make our roads safer, reduce congestion and improve productivity, given that urban Indians, on an average, spend 1.5 hours more in daily traffic, as compared to those in neighbouring Asian countries. On the other hand, they can be potentially overwhelmed in highly congested driving scenarios in the country and fail to function well in the absence of the infrastructural overhaul that is required to normalize their existence.

Whether we like it or not AV is the next evolutionary step for cars.Back in 2018, Mahindra & Mahindra showcased an autonomous tractor, stating that it intends to develop fully autonomous tractors in the future. Making a self-driving car in India requires several million km of real-world data, based on which an algorithm suited to Indian driving conditions can be created. At present no Indian automaker has announced plans of deploying AI tech anytime soon.

An AV like any technology has its defects, A self-driving vehicle could suffer from both computer and physical defects .Additionally , operating in hazardous weather conditions could lead to many of the same problems that happen today.Heavy rain, fog, or snow could impair a self-driving vehicles sensor.

3.7 Autonomous drones (UAV-unmanned aerial vehicles)

In the event of an emergency, autonomous drones provide immediate situational awareness and help direct response efforts without an on-site pilot. Port & terminal operators across the globe leverage on-site autonomous industrial drone systems to overcome a wide variety of security and inspection challenges.

AI can be used to automate the control of drones, including their navigation and movement. This can be done using a variety of methods, including GPS tracking, computer vision, and machine learning algorithms. Military autonomous drones (UAVs) can fly to a specific location, pick their own targets and kill without the assistance of a remote human operator.

Lethal autonomous weapon systems (LAWS) are a special class of weapon systems that use sensor suites and computer algorithms to independently identify a target and employ an onboard weapon system to engage and destroy the target without manual human control of the system.

Limited flight endurance and payload capacity

Further it requires recharging the battery, i.e. the lesser battery capacity is becoming one of the major issues. Another issue that is confronted by autonomous drone development is the load capacity. A drone can handle only five pounds at the max.

3.8 Unmanned ground Vehicle

Semi-autonomous and autonomous vehicles offer the opportunity to significantly reduce the number of troops required to conduct a convoy. The Army is developing leader-follower technology, which allows a manned lead vehicle to travel along a route and have some semi-autonomous vehicles following along in the sequence.

The Indian Army presently does not have any ground vehicle to undertake unmanned autonomous capturing of intelligence, surveillance and delivery of loads / casualty evacuation.

The THMIS (Tracked Hybrid Modular Infantry System), unmanned ground vehicle (UGV), is a ground-based armed drone vehicle designed largely for military applications and is built by Milrem Robotics in Estonia.

Building a full working model of an unmanned ground vehicle using Convolutional Neural Networks (CNN) is a complex task that requires expertise in several areas, including robotics, computer vision, and deep learning. However, I can provide you with a high-level overview of the process involved:

- Data collection: Collecting a dataset of images and corresponding labels that the CNN will use to learn to identify objects and navigate the environment.
- Data preprocessing: Preprocessing the data to ensure that it is of high quality and consistency, which may include tasks such as image cropping, resizing, and normalization.
- Model architecture: Designing the CNN architecture, which will typically consist of multiple layers of convolution, pooling, and fully connected layers.

- Model training: Training the CNN using the collected and preprocessed data. This involves optimizing the model parameters to minimize the error between the predicted output and the ground truth.
- Evaluation: Evaluating the performance of the model on a separate test set to ensure that it can generalize to new, unseen data.
- Deployment: Deploying the trained model to the unmanned ground vehicle and integrating it with the necessary hardware and software components, such as sensors, actuators, and communication modules.
- Testing and fine-tuning: Testing the unmanned ground vehicle in various environments and fine-tuning the model parameters to optimize its performance.

CHAPTER 4

FABRICATION OF CHASSIS

4.1. INTRODUCTION

Overall, fabricating a chassis using laser cutting and high-strength steel requires expertise in several areas, including design, materials selection, laser cutting, forming, welding, and finishing. It is a complex process that requires careful planning and execution to ensure a high-quality and durable final chassis.

4.2. DESIGN OF THE CHASSIS

Overall, designing an unmanned ground vehicle for military purposes requires careful consideration of these key design features to ensure its durability, reliability, and ability to operate in complex and hazardous environments.

1. Design: The first step in fabricating a chassis is to design the chassis using **Autodesk fusion 360 software**. This will include determining the dimensions, shape, and layout of the chassis, as well as any necessary cut outs, holes, or brackets.
2. Robust Chassis: The UGV should have a robust chassis that can withstand high impacts and rough terrain. The chassis should be made of lightweight, high-strength materials, such as aluminum or carbon fiber, and incorporate shock absorbers to minimize vibrations and impacts. Additionally, the chassis should be designed to protect the internal components from external damage.
3. Power Source: The UGV will need a power source that can provide sufficient power to move the vehicle and operate its various systems. Depending on the size and weight of the vehicle, the power source may be a battery, fuel cell, or combustion engine.
4. Sensors and Communications: The UGV should be equipped with sensors such as cameras, lidar, and GPS to enable the vehicle to detect obstacles and navigate its surroundings. It should also have a robust communication system, such as satellite or radio, to ensure reliable communication with the remote control operator.
5. Autonomous Navigation: The UGV should be able to operate autonomously and navigate its environment without human intervention. This may require

advanced sensors and algorithms to enable the vehicle to detect and avoid obstacles, and navigate complex terrain.

6. Durability: The UGV should be designed to withstand extreme environmental conditions, such as extreme temperatures, dust, and moisture. It should also be able to withstand damage from small arms fire and shrapnel.
7. Armament: The UGV may be designed to carry various types of weapons, such as machine guns or grenade launchers, to provide support to ground troops in combat situations.
8. Kevlar tracks: The UGV may be equipped with Kevlar tracks to provide maximum traction and durability in rough terrain.

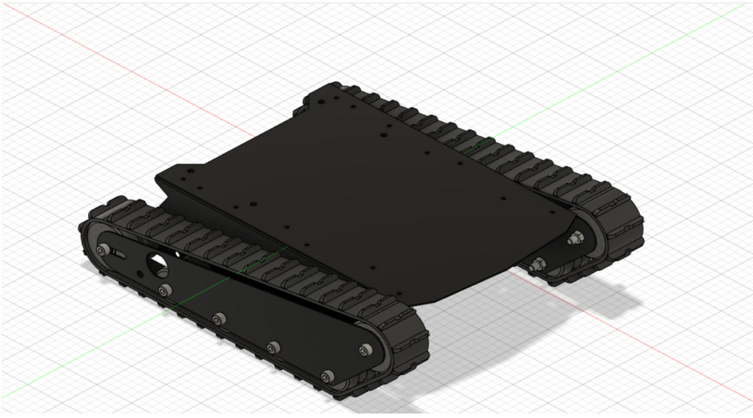

Figure 4.1 Design of the chassis

Entity	Value
Length	460 mm
Width	374 mm
Height	160 mm
Weight (chassis)	
Aspect ratio	1.22
Front area (chassis)	19650 mm^2
Frontal area (total estimated)	48296 mm^2
Material used (chassis)	Mild steel
Density	0.008 g/mm^3

Table 4.1 Specifications of chassis

4.3. STEPS INVOLVED IN THE FABRICATION PROCESS

1. Material selection: Selecting the appropriate high-strength steel for the chassis, which will depend on factors such as the expected loads, stresses, and operating conditions. Some common high-strength steels used for chassis fabrication include S690QL, S960QL, and Domex.
2. Laser cutting: Laser cutting involves using a laser beam to cut the steel sheet or plate to the desired shape and size. This provides precise and accurate cuts that can be repeated with minimal variation.
3. Forming: After laser cutting, the steel sheet or plate is formed into the desired shape using various forming techniques such as bending, rolling, or stamping. This will depend on the design and function of the chassis.
4. Welding: The various components of the chassis are welded together using high-strength welding techniques such as gas metal arc welding (GMAW) or tungsten inert gas (TIG) welding. This creates a strong and durable bond between the steel components.
5. Finishing: After welding, the chassis is finished by cleaning, painting, or powder-coating. This will help to protect the steel from corrosion and improve its aesthetic appearance.

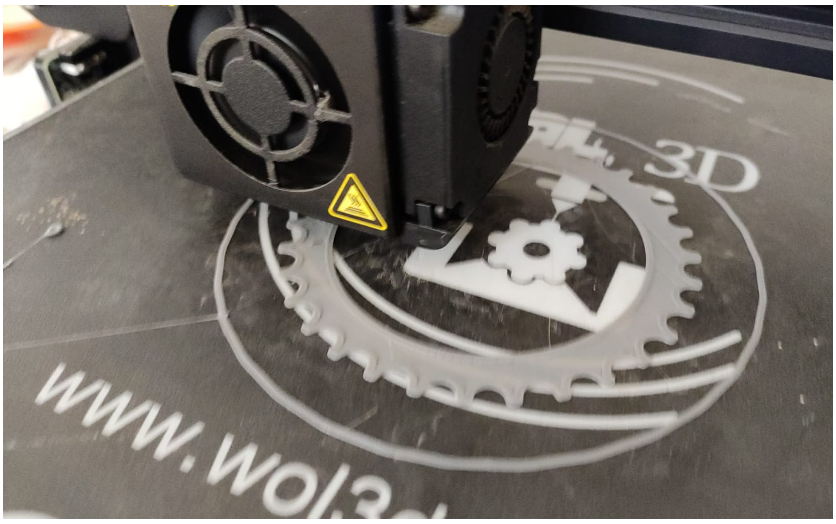

Figure 4.2 Fabricated model of the chassis

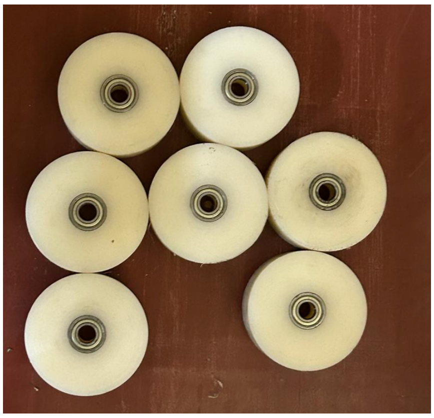

Figure 4.3 Fabricated model of the chassis.

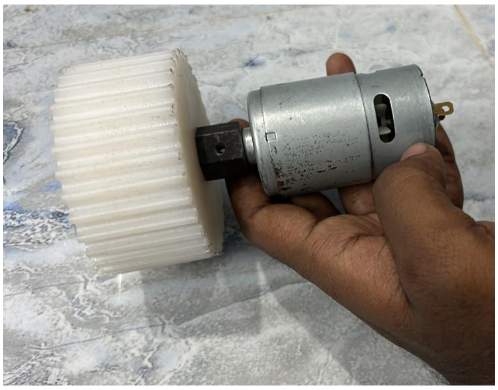

Figure 4.4 Fabricated model of the chassis

Figure 4.5 Fabricated model of the chassis

Figure 4.6 Fabricated model of the chassis

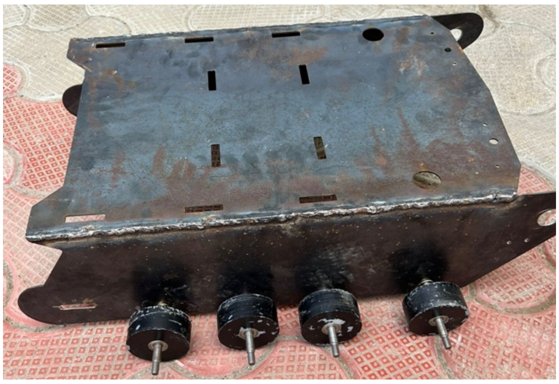

Figure 4.7 Fabricated model of the chassis

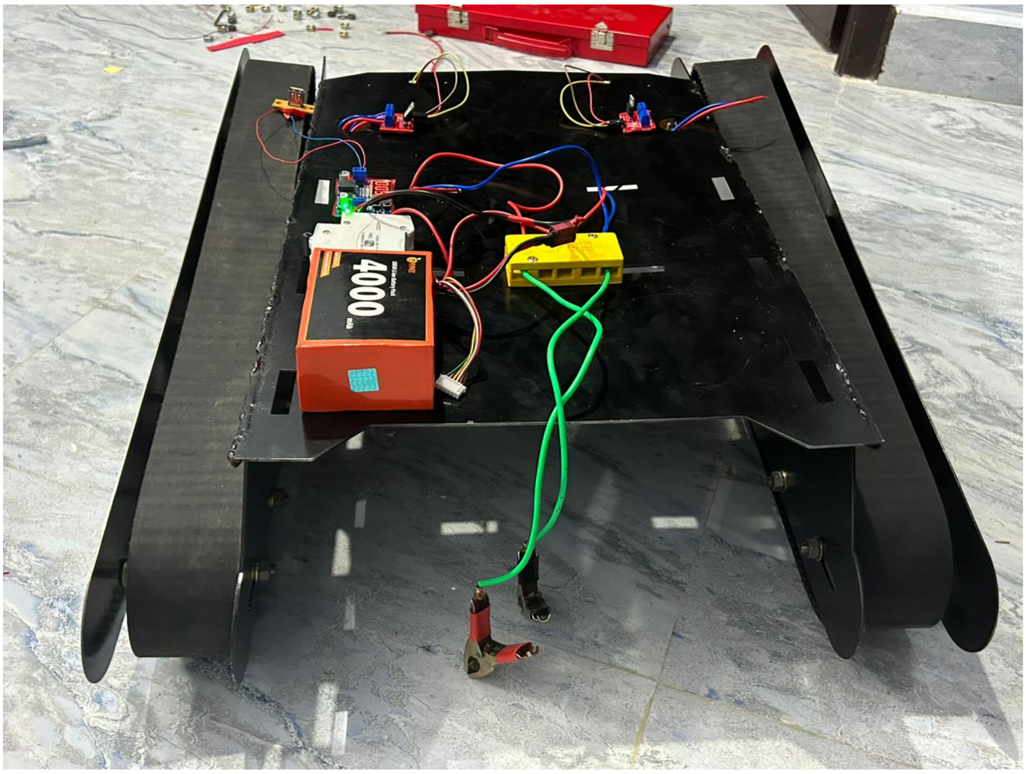

Figure 4.8 Fabricated model of the chassis

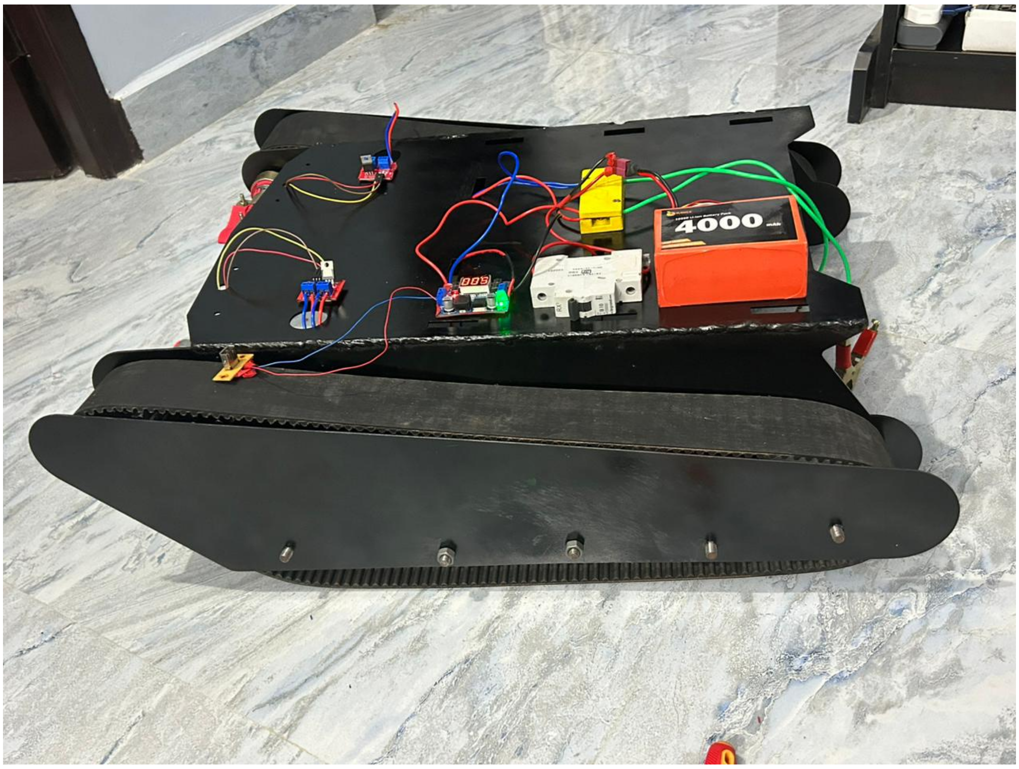

Figure 4.9 Fabricated model of the chassis

4.4. MATERIALS USED FOR THE FABRICATION

The use of high-strength steel and Kevlar tracks can provide several advantages for the design of an unmanned ground vehicle (UGV) for military purposes.

High-strength steel is a popular material for UGV chassis design because of its strength and durability. It can withstand high impacts and rough terrain, making it an ideal material for use in military applications where UGVs may be exposed to harsh environments and challenging terrain. High-strength steel can also provide a lightweight and cost-effective solution for UGV chassis design, which is essential for improving the vehicle's mobility and reducing its overall weight.

Kevlar tracks are also an excellent choice for UGV mobility because of their durability and resistance to damage. Kevlar is a strong synthetic material that can withstand high levels of stress and impact, making it ideal for use in UGV tracks. The Kevlar tracks can provide a high level of traction and stability, allowing the UGV to operate effectively in various types of terrain, including soft soil, rocks, and steep inclines. Kevlar tracks can also provide protection against damage from small arms fire, shrapnel, and explosive devices, which is essential for UGVs operating in hazardous environments.

Overall, the use of high-strength steel for the UGV chassis and Kevlar tracks for mobility can provide several advantages, including improved durability, mobility, and protection against damage. These design features can help to enhance the effectiveness of UGVs in military applications, allowing them to operate effectively in challenging and hazardous environments.

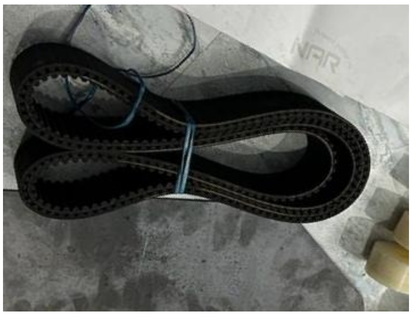

Figure 4.10 Kevlar Tracks.

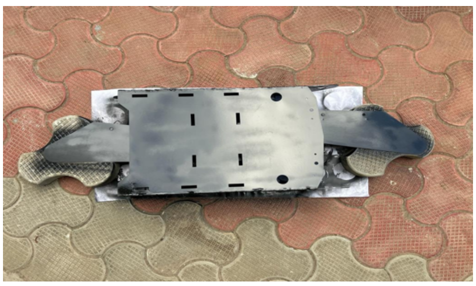

FIGURE 4.11 HIGH STRENGTH STEEL PANELS FOR THE CHASSIS

CHAPTER 5

ELECTRONICS AND ELECTRICALS

5.1.RASPBERRY PI:

Raspberry Pi 4 4Gb is the latest product in the popular Raspberry Pi range of computers. It offers ground-breaking increases in processor speed, multimedia performance, memory, and connectivity compared to the prior-generation Raspberry Pi 4 4 Gb key features include a high-performance 64-bit quad-core processor, dual-display support at resolutions up to 4K via a pair of micro-HDMI ports, hardware video decode at up to 4Kp60, up to 4GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE capability (via a separate PoE HAT add-on). The dual-band wireless LAN and Bluetooth have modular compliance certification, allowing the board to be designed into end products with significantly reduced compliance testing, improving both cost and time to market.

SPECIFICATION	INFERENCE
Processor	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory	1GB, 2GB or 4GB LPDDR4
Connectivity	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports
Input power	5V DC via USB-C connector (minimum 3A1) 5V DC via GPIO header (minimum 3A1) Power over Ethernet (PoE)–enabled (requires separate PoE HAT)

Table 5.1 Specifications of Raspberry Pi 4

5.2. IRF-520 MOSFET CONTROLLER

RF520 MOSFET Driver Module is an easy hookup to 3-5v microcontroller. Pulse current through a PWM pin, and use this to control a pump, LED, or other DC device at up to 24v, and ~5A. Its main purpose of this IRF520 MOSFET Driver Module is to provide a low-cost way to drive a DC motor for robotics applications, but the module can be used to control most high current DC loads. Screw terminals are provided to interface with your load an external power source. An LED indicator provides a visual indication of when your load is being switched. You connect the power for the device you want to control the Vin and GND. You connect your device to the V+ and V- screw terminals. When the SIG line is HIGH, a little LED lights up, and the GND is ‘connected’ to the V-. When the SIG line is LOW, the LED turns off, and the GND becomes ‘disconnected’ to the V-. The V+ and Vin are continuously connected. You can power this device from your Arduino, VCC goes to the 5V connection and GND to the GND pin on your Arduino.IRF520 MOSFET Driver Module is a breakout board for the IFR520 MOSFET transistor. The module is designed to switch heavy DC loads from a single digital pin of your microcontroller.

SPECIFICATION	VALUE
Operating Voltage(VDC)	3.3 to 5
Port	Digital Level
Output load voltage(V)	0 to 24
Output load current(A)	<5 (1A above need to add heat sink)
Platform	Arduino, MCU, ARM, Raspberry PI

0 Table 5.2 Specifications of IRF-520

5.3. 10A DC-DC CONVERTER

The 10A DC-DC Step-down Adjustable Constant Voltage Module can be used to get adjustable output voltage ranges from 1.5V to 35V. The module provides a wide range of current output up to 10 A. With a heat sink mounted it can easily manage to run a high power application continuously (provided that for continuous high power output you need to use a cooling fan on the heat sink).

The Module is made from a dedicated benchmark IC and high-precision current sensing resistor, providing a more stable constant current, (when 20°C to 100°C constant current 1A, temperature drift less than 1%); and therefore it is particularly most suitable for LED driver applications. The output current of the module can be raised up to 10A(for continuous use it is recommended up to 8A with an additional cooling fan).To enhance the stabilization in output voltage it has four high-frequency capacitance which gives the lower output ripple. Double heat sink design provides easy and fast heat dissipation. MOS Schottky diode independent heat sink, which heat dissipation is good, and won’t affect each other. Also using large size Sendust Core and double pure copper wiring, improve working efficiency, reduce fever.The module has onboard 3296 multi turn potentiometers for high accuracy voltage and current regulation to provide good stability. As voltage and current are adjustable, it gives very easy to use at multiple Applications such as battery charging, LED Driver power supply, Vehicle Power Supply, etc which makes the module Multipurpose And hence this module is always in demand from our maker because of its wide range of application and low cost.

5.4. CSI CAMERA MODULE

Commonly used in embedded vision systems, MIPI CSI-2 is a camera interface that connects an image sensor with an embedded board to control and process the image data. This helps the sensor and embedded board to act together as a camera system to capture images .The Camera Serial Interface (CSI) is a specification of the Mobile Industry Processor Interface (MIPI) Alliance. It defines an interface between a camera and a host processor. The camera module is a product used to take photos and videos from mobile devices, such as smartphones, automobiles, and smart home appliances .

5.5. IR SENSOR:

IR sensor is an electronic device that emits light in order to sense some object of the surroundings. An IR sensor can measure the heat of an object as well as detect the motion. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation. These types of radiation are invisible to our eyes, but infrared sensors can detect these radiations .The emitter is simply an IR LED and the detector is simply an IR photodiode . Photodiodes are sensitive to IR light of the same wavelength which is emitted by the IR LED. When IR light falls on the photodiode, the resistances and the output voltages will change in proportion to the magnitude of the IR light received .There are five basic elements used in a typical infrared detection system: an infrared source, a transmission medium, optical component, infrared detectors or receivers and signal processing. Infrared lasers and Infrared LED's of specific wavelength used as infrared sources .The three main types of media used for infrared transmission are vacuum, atmosphere and optical fibers. Optical components are used to focus the infrared radiation or to limit the spectral response.

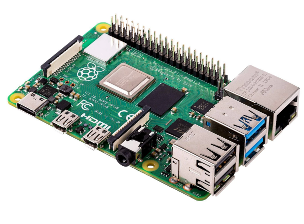

Figure 5.1 Raspberry Pi 4 4GB

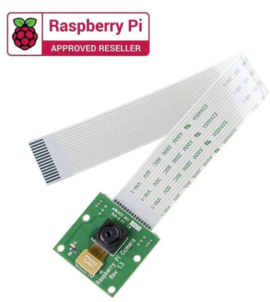

Figure 5.2 CSI Camera Module

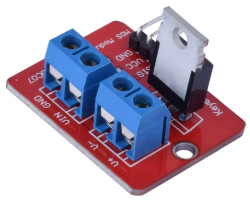

Figure 5.3 IRF520 MOSFET controller

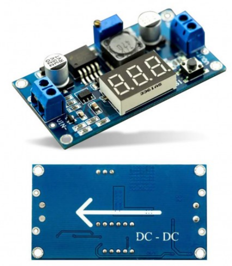

Figure 5.4 DC to DC Converter

5.6. DC MOTOR

DC motors are highly versatile, and are used in many appliances and technologies. Whether you're a hobbyist or industrial manufacturer, this DC electric motor can be used in both model and industrial applications. Typical applications include;

- Prototyping
- Water pumps
- Camera Systems
- Robotics

PARAMETERS	SYMBOLS	VALUES	UNITS
Length of the vehicle	M	10	Kg
Wheel Radius	R	0.05	m
Coefficient of rolling resistance	µr	0.01	-
Density of air	P	1.225	Kg/m2
Frontal Area	A	0.6	m2
Coefficient of Drag	Cd	0.6	-

5.6.1. TRACTIVE EFFORT:

Aerodynamic drag force, Fad = 1/2* $CdA\rho v2$
= 0.5 x 0.6 x 0.095 x 1.225 x (4.167)2
= 0.607N
Rolling resistance,Fr = μr mgcosθ
= 0.01x 10 x 9.81
= 0.981N
Gradient resistance,Fg = mgsinθ
= 10 x 9.81 x sin 15
= 2.539N

5.6.2. POWER REQUIRED TO CRUISE THE VEHICLE AT 15 KMPH

When cruising the force acting are only rolling resistance and aerodynamic drag. Slope angle=0; acceleration=0

Total force = Fad + Frr= 0.607+ 0.981
$Ftotal$ = 1.588N
Torque at wheel = F total x R = 1.588 x 0.08 = 0.127Nm
Power required, $P = T *v$ /r= 0.127X 4.167/0.04
P = 13.23Watt
Efficiencies (η) =η motor x η wire = 0.95 x 0.9 = 0.85
Actual power required = 13.23/0.855 =15.47Watt

5.6.3. POWER REQUIRED WITH GRADE :

Slope angle=15
Total force = Fad + Frr+Fg= 0.607+ 0.981+2.539
$Ftotal$= 4.127N
Torque at wheel = F total x R = 4.127x 0.08 = 0.33016Nm
Power required, $P = T\ v\ /\ \ r$= 0.33016X 4.167/0.04
P = 34.4Watt
Efficiencies (η) =η motor x η wire = 0.95 x 0.9 = 0.855
Actual power required = 34.4/0.855 =40.23Watt

Figure 5.5 DC Motor

After calculating the required motor specifications we have obtained the Orange Motors with the following specifications .

SPECIFICATION	VALUE
Type	Planetary geared Dc motor
Input voltage (V)	12 V
Power (Watt)	60 Watts

Torque (Kg-cm)	22.5
Max rpm	800
Max current (A)	6

5.7. LITHIUM-ION BATTERY

A lithium-ion or Li-ion battery is a type of rechargeable battery which uses the reversible reduction of lithium ions to store energy. The anode (negative electrode) of a conventional lithium-ion cell is typically graphite made from carbon. The cathode (positive electrode) is typically a metal oxide. The electrolyte is typically a lithium salt in an organic solvent.

5.7.1. BATTERY CALCULATION:

Battery capacity: $C = (P \times T) \div (V \times \eta \times K)$
C–Battery capacity of the required configuration; unit–Amp × hour (Ah)
P–Load power; unit–Watt (W)
T–Backup time; unit–Hour (h)
V–Battery pack rated voltage; unit–Volt (v)
H–Battery inverter efficiency
K–Battery discharge coefficient
When the battery backup time is <3h, K = 0.6
$C = (40.39 \times 1) \div (12 \times 0.92 \times 0.6)$
C = 6.09AH

After calculating the desired battery capacity and power required for the prototype We have obtained the 18650 Li-ion Battery pack from Orange with the following specifications .

SPECIFICATION	VALUE
Nominal Capacity (mAh)	4000
Nominal Voltage (V)	18.5
Max. Charging Voltage (V)	22.5
Charging Cut-off Voltage (V)	22.5
Max. Charging Current (A)	4
Nominal Charge Current (A)	1.5
Cell configuration	5S-2P
Discharge rate	3C – 12 Amps

Table 5.3 Specifications of Lithium ion battery

5.8 BALANCE CHARGER:

This function checks the voltages of each cell in a battery pack and ensures that they all have the same voltage. This is a critical monitoring step in order to prevent damage to the battery itself as it allows the battery's voltages to remain even throughout. This makes it so that one or more cells don't under or over discharge, which can cause battery failure. Some batteries can also have built-in battery monitors and balancers, which you can check before picking a charger. Its function allows your battery to be charged faster, but it generally does not monitor or balance the voltage of individual cells. Instead, it will only look at the overall voltage. So, if the cells in your battery pack aren't balanced properly, it is possible to overcharge one or more cells. Make sure to pay attention to the state of your battery if you ever quick-charge it. Furthermore, the battery-charging process is non-linear, so most of the chemistry within the battery is capable of fast charging a battery from 0% to a little over 90% state of charge (SoC). This causes little damage to the battery itself. In the second half of a charge, the anodes become less receptive to lithium ions, so the charging rate greatly reduces. If the charging rate exceeds the ability of lithium ions to embed into negative graphite electrodes, lithium deposits can form, which then can cause a battery to short circuit or even explode. This makes it even more vital that you ensure your RC battery cells are capable of accepting fast charging before you actually utilize the function.

Figure 5.6 Lithium battery with balanced charger

CHAPTER 6

DEVOLOPMENT OF SOFTWARE

6.1. INTRODUCTION

Machine learning is a particular class of software algorithm. It’s basically a way of instructing a computer to learn by feeding it an enormous amount of data and telling it when it’s doing better or worse. The machine-learning algorithm modifies itself to do better more often.

6.2. REQUIREMNT OF ALGORITHMS

1. To detect off road path along with its curvature
2. To detect and classify various objects
3. To measure the dimensions of the object using monocular vision

6.3. PATH DETECTION USING EDGE OPERATORS

There are many different edge detection methods out there and if you ever wondered how they compare with each other then you came to the right place, so let’s compare them .We will be implementing some of the most commonly used methods and also using methods from OpenCV and PIL

Various operators tested are :

- Sobel edge detector
- Prewitt edge detector
- Laplacian edge detector
- Canny edge detector

6.4. TO DETECT AND CLASSIFY OBJECTS

*A single convolutional neural network simultaneously predicts multiple **bounding boxes** and **class probabilities** for those boxes* .YOLO uses features from the entire image to predict each bounding box and their classes which it does **simultaneously**. Similar to humans, YOLO can pretty much immediately recognize where and what objects are within a given image. When running on an image, YOLO first divides the image into an **S by S** grid. Within each grid cell, YOLO will predict the **locations**, **sizes**, and **confidence scores** of the predetermined number of **bounding boxes**- essentially predicting the class and potential place where an object can be. If the centre of an object falls into the grid cell, then the bounding boxes of that grid cell are responsible for accurately locating and predicting that object. YOLO locates and classifies objects within an image/video. **Object Detection** algorithms like YOLO, combined with the many other sensors on a self-driving car like Li-Dar, allow us to build fully autonomous cars that can drive faster, safer, and better than any human can. If you are interested in diving deeper into self-driving cars

6.5 DIMENSION DETECTION USING MONOCULAR VISION

In Order To Determine The Size Of An Object In An Image, We First Need To Perform A “Calibration” Using A Reference Object. Our Reference Object Should Have Two Important Properties:

Property #1: We Should Know The Dimensions Of This Object (In Terms Of Width Or Height) In A Measurable.

Property #2: We Should Be Able To Easily Find This Reference Object In An Image, Either Based On The Placement Of The Object (Such As The Reference Object Always Being Placed In The Top-Left Corner Of An Image) Or Via Appearances (Like Being A Distinctive colour Or Shape, Unique And Different From All Other Objects In The Image). In Either Case, Our Reference Should Be Uniquely Identifiable In Some Manner.

The captured image is used to detect the dimensions of various objects present in the scenario using a reference object which is any part of the chassis specially used for the particular purpose .

6.6. TO CONTROL MOTORS USING MOSFET CONTROLLER

Precise control of motor speed and direction of rotation is achieved by generating PWM signals using python RPI.GPIO library in raspberry pi GPIO pins . Based on the PWM signals given to IRF-520 MOSFET driver the output voltage to the motor is varied . The transmitting frequency of the PWM signals is set to 55Hz and the control values ranges from 0-100 representing 0 for off state and 100 for maximum power. A map for a range of PWM signals and motor rpm has be experimentally plotted and used as test data for the algorithm.

6.7. VIDEO STREAMING AND EMERGENCY STOP

Python socket library is used to steam live video relay to nearby connected devices and implement emergency stop function remotely . Both the raspberry pi and the remote device has to be connected to the same Wireless Local Network to establish live steam .

6.8. PYTHON PROGRAM

```
import RPi.GPIO as GPIO
from time import sleep
import numpy as np
import utlis
import cv2
import socket,pickle,struct



class motor():

    def __init__(self):
        self.p1=11
        self.p2=12
        GPIO.setmode(GPIO.BOARD)
        GPIO.setup(self.p1,GPIO.OUT)
        GPIO.setup(self.p2,GPIO.OUT)
        self.motr=GPIO.PWM(self.p1,55)
        self.motl=GPIO.PWM(self.p2,55)

    def move(self,speed=0):
        self.motr.start(speed)
        self.motl.start(speed)
    def stop(self):
        self.motr.ChangeDutyCycle(0)
        self.motl.ChangeDutyCycle(0)
```

```
    def direction(self,t):
        if t>0:
            temp=t.round(2)
            tempr=t*10
            templ=0
        else:
            temp=abs(t)
            temp=t.round(2)
            templ=t*10
            tempr=0

        self.motr.ChangeDutyCycle(tempr)
        self.motl.ChangeDutyCycle(templ)

def thresholding(img):
    imgHsv = cv2.cvtColor(img,cv2.COLOR_BGR2HSV)
    lowerWhite = np.array([80,0,0])
    upperWhite = np.array([255,160,255])
    maskWhite = cv2.inRange(imgHsv,lowerWhite,upperWhite)
    return maskWhite

def warpImg(img,points,w,h,inv = False):
    pts1 = np.float32(points)
    pts2 = np.float32([[0,0],[w,0],[0,h],[w,h]])
    if inv:
        matrix = cv2.getPerspectiveTransform(pts2, pts1)
    else:
        matrix = cv2.getPerspectiveTransform(pts1,pts2)
    imgWarp = cv2.warpPerspective(img,matrix,(w,h))
    return imgWarp

def nothing(a):
    pass

def initializeTrackbars(intialTracbarVals,wT=480, hT=240):
    cv2.namedWindow("Trackbars")
    cv2.resizeWindow("Trackbars", 360, 240)
    cv2.createTrackbar("Width Top", "Trackbars", intialTracbarVals[0],wT//2, nothing)
    cv2.createTrackbar("Height Top", "Trackbars", intialTracbarVals[1], hT, nothing)
    cv2.createTrackbar("Width Bottom", "Trackbars", intialTracbarVals[2],wT//2,
nothing)
    cv2.createTrackbar("Height Bottom", "Trackbars", intialTracbarVals[3], hT, nothing)

def valTrackbars(wT=480, hT=240):
    widthTop = cv2.getTrackbarPos("Width Top", "Trackbars")
    heightTop = cv2.getTrackbarPos("Height Top", "Trackbars")
    widthBottom = cv2.getTrackbarPos("Width Bottom", "Trackbars")
    heightBottom = cv2.getTrackbarPos("Height Bottom", "Trackbars")
    points = np.float32([(widthTop, heightTop), (wT-widthTop, heightTop),
                      (widthBottom , heightBottom ), (wT-widthBottom, heightBottom)])
    return points

def drawPoints(img,points):
```

```
    for x in range(4):
        cv2.circle(img,(int(points[x][0]),int(points[x][1])),15,(0,0,255),cv2.FILLED)
    return img

def getHistogram(img,minPer=0.1,display= False,region=1):

    if region ==1:
        histValues = np.sum(img, axis=0)
    else:
        histValues = np.sum(img[img.shape[0]//region:,:], axis=0)

    #print(histValues)
    maxValue = np.max(histValues)
    minValue = minPer*maxValue

    indexArray = np.where(histValues >= minValue)
    basePoint = int(np.average(indexArray))
    #print(basePoint)

    if display:
        imgHist = np.zeros((img.shape[0],img.shape[1],3),np.uint8)
        for x,intensity in enumerate(histValues):
            cv2.line(imgHist,(x,img.shape[0]),(x,img.shape[0]-
intensity//255//region),(255,0,255),1)
            cv2.circle(imgHist,(basePoint,img.shape[0]),20,(0,255,255),cv2.FILLED)
        return basePoint,imgHist

    return basePoint

def stackImages(scale,imgArray):
    rows = len(imgArray)
    cols = len(imgArray[0])
    rowsAvailable = isinstance(imgArray[0], list)
    width = imgArray[0][0].shape[1]
    height = imgArray[0][0].shape[0]
    if rowsAvailable:
        for x in range ( 0, rows):
            for y in range(0, cols):
                if imgArray[x][y].shape[:2] == imgArray[0][0].shape [:2]:
                    imgArray[x][y] = cv2.resize(imgArray[x][y], (0, 0), None, scale,
scale)
                else:
                    imgArray[x][y] = cv2.resize(imgArray[x][y],
(imgArray[0][0].shape[1], imgArray[0][0].shape[0]), None, scale, scale)
                if len(imgArray[x][y].shape) == 2: imgArray[x][y]= cv2.cvtColor(
imgArray[x][y], cv2.COLOR_GRAY2BGR)
        imageBlank = np.zeros((height, width, 3), np.uint8)
        hor = [imageBlank]*rows
        hor_con = [imageBlank]*rows
        for x in range(0, rows):
            hor[x] = np.hstack(imgArray[x])
        ver = np.vstack(hor)
    else:
```

```
        for x in range(0, rows):
            if imgArray[x].shape[:2] == imgArray[0].shape[:2]:
                imgArray[x] = cv2.resize(imgArray[x], (0, 0), None, scale, scale)
            else:
                imgArray[x] = cv2.resize(imgArray[x], (imgArray[0].shape[1],
imgArray[0].shape[0]), None,scale, scale)
            if len(imgArray[x].shape) == 2: imgArray[x] = cv2.cvtColor(imgArray[x],
cv2.COLOR_GRAY2BGR)
        hor= np.hstack(imgArray)
        ver = hor
    return ver


curveList = []
avgVal=10

def getLaneCurve(img,display=2):

    imgCopy = img.copy()
    imgResult = img.copy()
    imgThres = utlis.thresholding(img)

    hT, wT, c = img.shape
    points = utlis.valTrackbars()
    imgWarp = utlis.warpImg(imgThres,points,wT,hT)
    imgWarpPoints = utlis.drawPoints(imgCopy,points)

    middlePoint,imgHist = utlis.getHistogram(imgWarp,display=True,minPer=0.5,region=4)
    curveAveragePoint, imgHist = utlis.getHistogram(imgWarp, display=True, minPer=0.9)
    curveRaw = curveAveragePoint - middlePoint

    curveList.append(curveRaw)
    if len(curveList)>avgVal:
        curveList.pop(0)
    curve = int(sum(curveList)/len(curveList))

    if display != 0:
        imgInvWarp = utlis.warpImg(imgWarp, points, wT, hT, inv=True)
        imgInvWarp = cv2.cvtColor(imgInvWarp, cv2.COLOR_GRAY2BGR)
        imgInvWarp[0:hT // 3, 0:wT] = 0, 0, 0
        imgLaneColor = np.zeros_like(img)
        imgLaneColor[:] = 0, 255, 0
        imgLaneColor = cv2.bitwise_and(imgInvWarp, imgLaneColor)
        imgResult = cv2.addWeighted(imgResult, 1, imgLaneColor, 1, 0)
        midY = 450
        cv2.putText(imgResult, str(curve), (wT // 2 - 80, 85),
cv2.FONT_HERSHEY_COMPLEX, 2, (255, 0, 255), 3)
        cv2.line(imgResult, (wT // 2, midY),(wT // 2 + (curve * 3), midY), (255, 0,
255), 5)
        cv2.line(imgResult, ((wT // 2 + (curve * 3)), midY - 25), (wT // 2 + (curve *
3), midY + 25), (0, 255, 0), 5)
        for x in range(-30, 30):
            w = wT // 20
```

```
            cv2.line(imgResult, (w * x + int(curve // 50), midY - 10),
                     (w * x + int(curve // 50), midY + 10), (0, 0, 255), 2)
    if display == 2:
        imgStacked = utlis.stackImages(0.7, ([img, imgWarpPoints, imgWarp],
                                             [imgHist, imgLaneColor, imgResult]))
        cv2.imshow('ImageStack', imgStacked)
    elif display == 1:
        cv2.imshow('Resutlt', imgResult)

    #### NORMALIZATION
    curve = curve/100
    if curve>1: curve ==1
    if curve<-1:curve == -1

    return curve

vehiclecontrol=motor()
cap = cv2.VideoCapture(0)
cap.set(3, 1280)
cap.set(4, 720)
cap.set(10, 70)

server_socket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
host_name  = socket.gethostname()
host_ip = socket.gethostbyname(host_name)
print('HOST IP:',host_ip)
port = 9999
socket_address = (host_ip,port)
server_socket.bind(socket_address)

server_socket.listen(5)

classNames = []
classFile ='coco.names'
warning=["person","vase","bus","laptop","remote","cell
phone","dog","banana","cat","stop sign"]
with open(classFile,'rt') as f:
    classNames=[line.rstrip() for line in f]


    configPath = 'ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
    weightsPath = 'frozen_inference_graph.pb'

    net = cv2.dnn_DetectionModel(weightsPath, configPath)
    net.setInputSize(320, 320)
    net.setInputScale(1.0 / 127.5)
    net.setInputMean((127.5, 127.5, 127.5))
    net.setInputSwapRB(True)
    success=True

    while success:
        vehiclecontrol.move(100)
        sleep(1)
```

```
        success, img = cap.read()
        classIds, confs, bbox = net.detect(img, confThreshold=0.50)
        print(classIds, bbox)
        if len(classIds)!=0:
            for classId, confidence, box in zip(classIds.flatten(), confs.flatten(),
bbox):
                if(classNames[classId - 1] in warning):
                     cv2.rectangle(img, box, color=(0,0,255), thickness=2)
                     cv2.putText(img,classNames[classId -
1].upper(),(box[0]+10,box[1]+30),
                            cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 255), 2)
                     cv2.putText(img, str(round(confidence * 100, 2)), (box[0]+200,
box[1]+30),
                            cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0,255), 2)
                     vehiclecontrol.stop()

                else:
                    cv2.rectangle(img, box, color=(0, 255, 0), thickness=2)
                    cv2.putText(img,classNames[classId -
1].upper(),(box[0]+10,box[1]+30),
                            cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
                    cv2.putText(img, str(round(confidence * 100, 2)), (box[0]+200,
box[1]+30),
                            cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
                    frameCounter += 1
                    if cap.get(cv2.CAP_PROP_FRAME_COUNT) == frameCounter:
                        cap.set(cv2.CAP_PROP_POS_FRAMES, 0)
                        frameCounter = 0

                    success, img = cap.read()
                    img = cv2.resize(img,(480,240))
                    curve = getLaneCurve(img,display=2)
                    temp=curve*0.76+24
                    vehiclecontrol.direction(temp)
        cv2.imshow("Output", img)
        if(cv2.waitKey(1)==27):
            success=False
        client_socket,addr = server_socket.accept()

        if client_socket:
                vid = cv2.VideoCapture(0)

                while(vid.isOpened()):
                    img,frame = vid.read()
                    a = pickle.dumps(frame)
                    message = struct.pack("Q",len(a))+a
                    client_socket.sendall(message)

                    cv2.imshow('TRANSMITTING VIDEO',frame)
                    key = cv2.waitKey(1) & 0xFF
                    if key ==ord('q'):
                        client_socket.close()
```

CHAPTER 7

7.1 TESTING AND OPTIMIZATION OF PROTOTYPE

The accuracy of an machine learning model is highly influenced by the training and Testing data accumulated over a period of time . In order to improve accuracy, reliability and processing speed of the prototype is been tested with various edge detection algorithms , hardware components and video pre-processing techniques . The obtained results have been tabulated and compared in this chapter .

7.2 VALIDATION OF EDGE DETECTION ALGORITHMS

The efficiency of an edge detection algorithm can be characterized by the number of false or duplicate edges detected over real edges . This is also affected by vibration of camera and certain other parameters in order to count the edges detect and edge counting algorithm has been integrated .

7.3 LAPLACIAN VS CANNY EDGE DETECTION

Accuracy is the percentage of real edges detected over duplicate edge caused by video distortion Or even pre-processing methods.

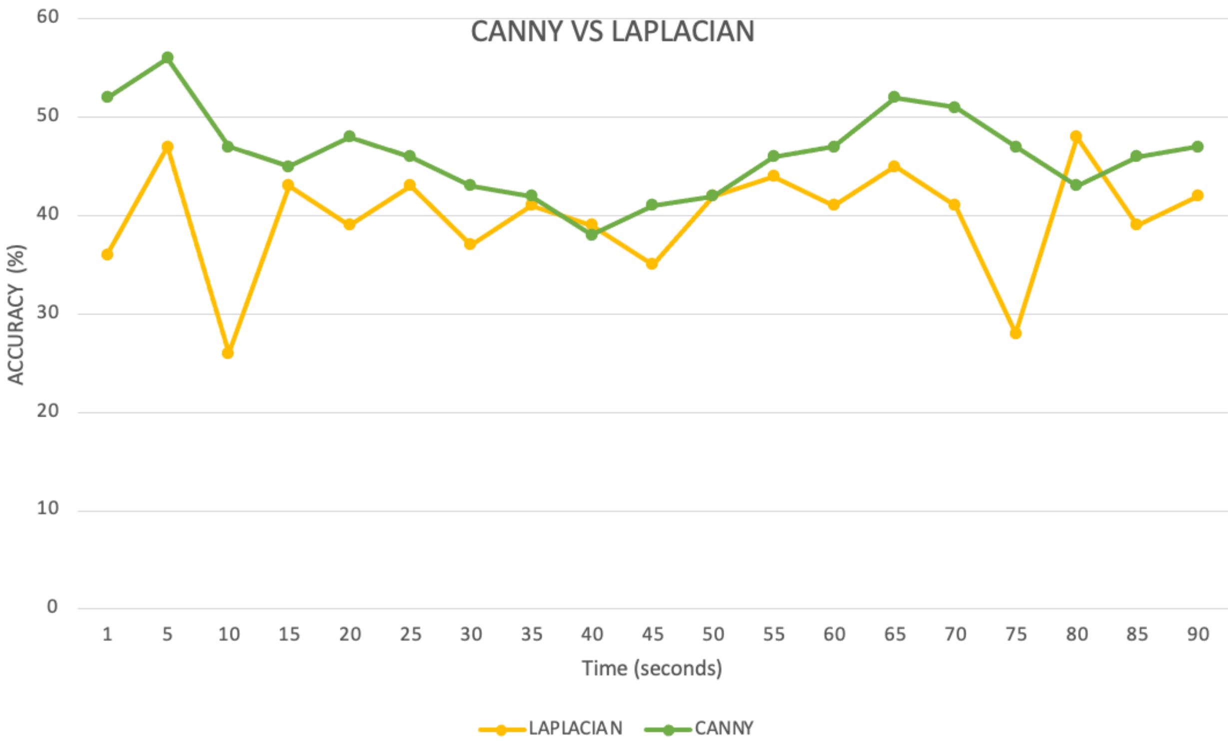

Figure 7.1 Comparison of canny vs Laplacian operators

It is inferred from the above plotted graph that the accuracy of canny edge detection operator is higher and higher precision compared to the Laplacian operator over a period of time . Hence the canny operator is used for detecting the off road path .

7.4 ACCURACY OF CANNY OPERATOR OVER THRESHOLD RANGE

Threshold value is the reference values used to detect the edges above or below a certain pixel value. Threshold value should be selected based on the lighting condition and different pre-processing methods . In general the number of edges detected is inversely proportional to the threshold value , for example lower threshold value would produces more edges regardless of it being edges of the off road track used for navigational purpose .

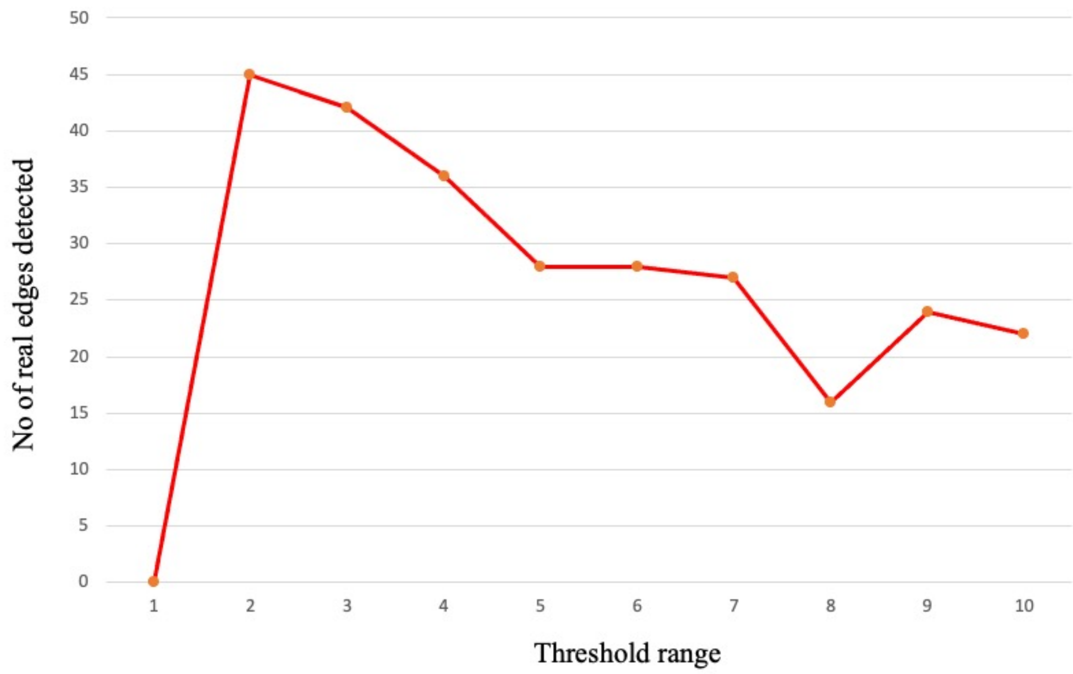

Figure 7.2 Graph for threshold value vs real edges detected

A threshold value of 2.27 is adopted as it produces the most number of real edges for our video pre-processing techniques and our testing environment .

7.5 COMPARSION OF USB-CAMERA VS CSI CAMERA

A csi-camera is dedicated camera setup for generic boards compared to general purpose usb-camera .

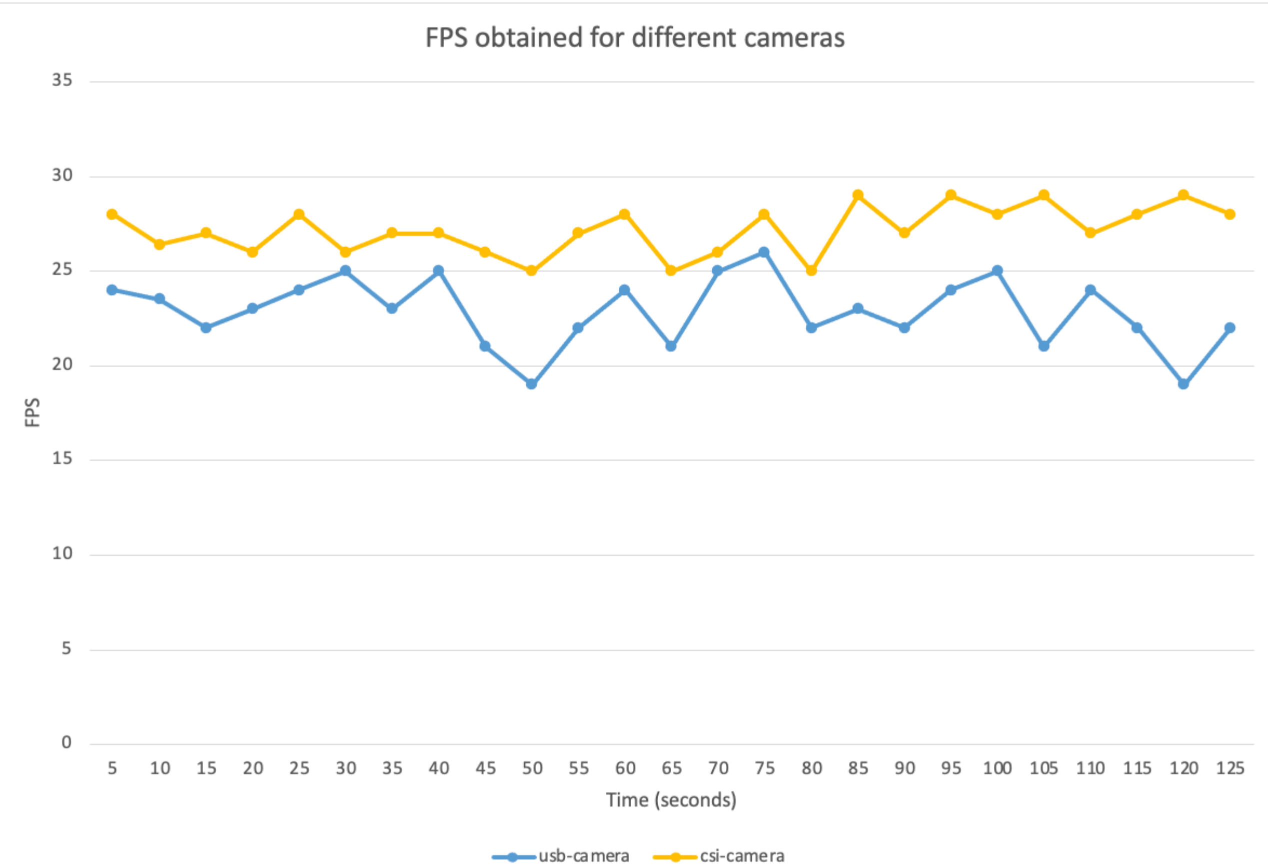

Figure 7.3 Graph for comparison of CSI vs USB camera

We have adopted to using csi-camera because of its ability to achieve higher frame rates in capturing video which would eventually decrease the reaction time of our prototype .

7.6 ACCURACY OF CERTAIN OBJECTS DETECTED USING YOLO ALGORITHM

We have selected a minimum accuracy percentage of 50 to classify objects below which we term the objects as un-identifiable after training and testing yolo for a certain period of time we were able to achieve better accuracy and precision for certain objects. It must also be noted that

Training and testing the machine learning model over a variety of data or in different environments would also result in better accuracy and precision over a period of time .

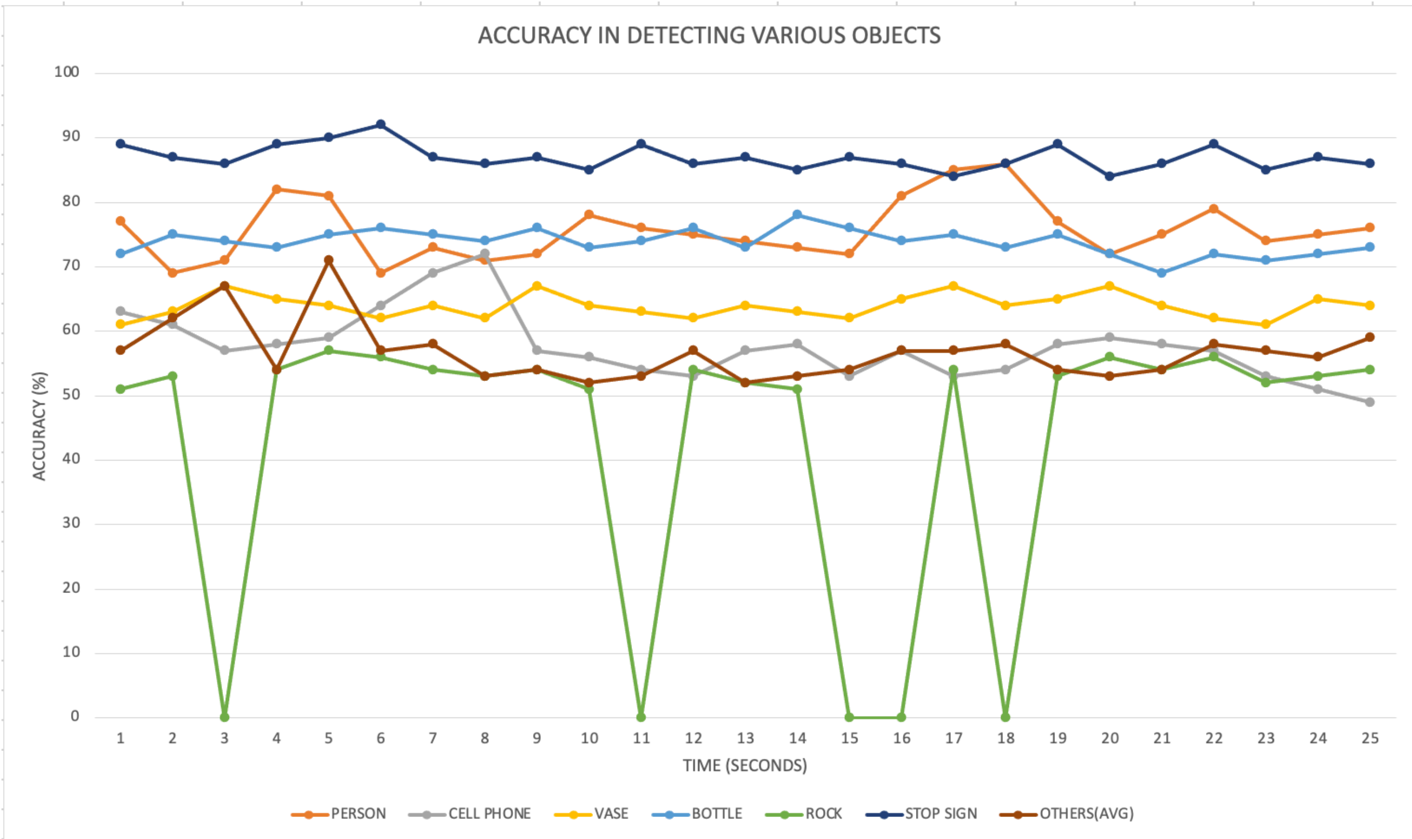

Figure 7.4 Graph for accuracy of various objects detected

7.7 MAPPING OF PWM SIGNALS WITH MOTOR RPM FOR PRECISE MOTOR CONTOL

Precise motor control can be achieved by experimentally collecting the motor rpm data for different PWM signals and plotting a graph which is given as a data map to the program for further control of motor in closed curves without over-steering or under-steering our prototype .

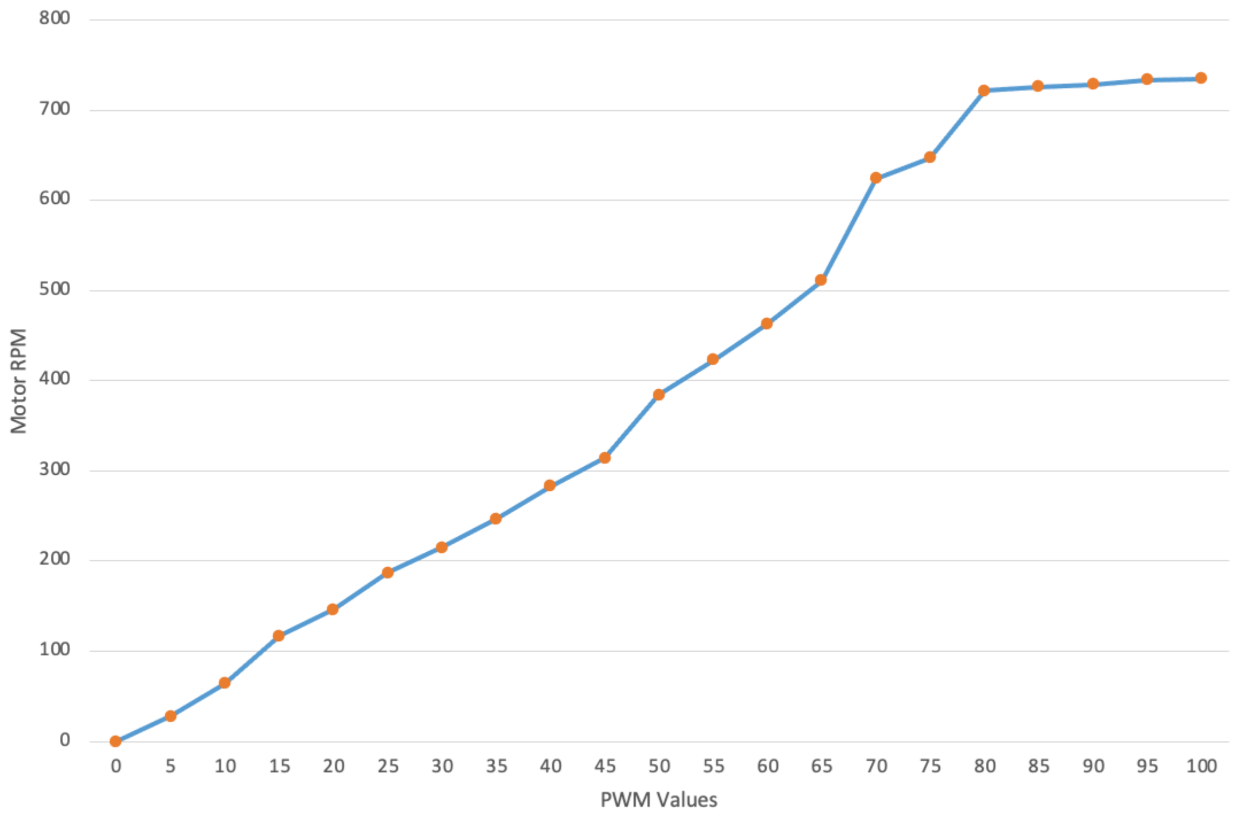

Figure 7.5 Graph for PWM signal vs Motor RPM

7.8 CONCLUSION

An autonomous vehicle is associated with a varied number of control variables and parameters of which testing of certain parameters have been conducted and improved over a period of time . Any autonomous prototype requires a lot of training and testing to be carried out before it can be deployed in an real environment .

CHAPTER 8

8.1 CONCLUSION

Our Objective is to design and develop a unmanned ground vehicle (UGV) for off road applications from scratch . In order to achieve our goal we have worked both on hardware and software level where we have developed our indigenous off road detection algorithm using canny edge detection and various image pre-processing levels to achieve considerable reliability. To detect and classify various objects in our path we have used yolo (you only look once) algorithm .

We have designed our chassis using Autocad Fusion 360 and tested for maximum load carrying capacity and stress developed using ansys and with the results obtained we have optimized our design .

Manual calculations have been done to find the required motor power which 48 watts and battery capacity which is 6.09AH to power the prototype for considerable duration. Required electrical and electronics components have been purchased and tested for reliability .

We have completed the fabrication of our prototype using different raw materials and manufacturing techniques and tested in closed environment .

8.2 FUTURE WORKS

In order to achieve a higher degree of automation GPS functions can be integrated In software . A Gyroscope module can be integrated with electronics in order track the orientation of our prototype in an real environment and move accordingly .

Training and testing of the prototype in outdoor environment could be carried out in order to achieve better reliability .