Greenhouse monitoring and harvesting mobile robot with 6DOF manipulator utilizing ROS, Inverse Kinematics and deep learning models

by

Md. Hashibul Islam
17301194
Md. Firoz Wadud
17201118
Md. Raihan Rahman
17301197
A S M Hasibul Alam
18101273

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2022

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md. Hashibul Islam

Md. Hashibul Islam
17301194

Firoz Wadud

Md. Firoz Wadud
17201118

Md Raihan Rahman

Md. Raihan Rahman
17301197

A S M Hasibul Alam

A S M Hasibul Alam
18101273

Approval

The thesis/project titled Greenhouse monitoring and harvesting mobile robot with 6DOF manipulator utilizing ROS, Inverse Kinematics and deep learning models" submitted by

1. Md. Hashibul Islam (17301194)
2. Md. Firoz Wadud (17201118)
3. Md. Raihan Rahman (17301197)
4. A S M Hasibul Alam (18101273)

Of Fall, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on 20 January, 2022.

Examining Committee:

Supervisor: (Member)

Md. Khalilur Rhaman

Md. Khalilur Rhaman, PhD
Associate Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator
& Co-Supervisor: (Member)

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department: (Chair)

SHKazi

Sadia Hamid Kazi
Associate Professor
Department of Computer Science and Engineering
BRAC University

Abstract

The rapid climate change and scarcity of fertile land has been a global concern recently. To sustain the food supply its high time to think about the modern way of cultivating which is greenhouse. Taking these changes as well as The paradigm shift in people's occupation, we aim to build a Farming robot with the capability of monitoring and maintaining the soil and the farming environment. In addition this robot will be able to count the amount of vegetables and fruits and harvest them exactly when they are mature for consumption. To move forward with this goal in mind we have added a robotic arm of 6 degrees of freedom and wheel tracks for moving through the mud and soil. With the help of ROS gazebo and A* algorithm rover can make its path through the farm. For picking the vegetables, detecting any diseases on plants we have tried, compared and choose various state of art deep learning models. These models have been merged with object tracking and inverse kinematics algorithms for manipulating the end effector to desired point. Thus, we would have our automated farming robot. The combination of the technologies makes our robot different and effective than other farming robots. As the components used in this robot are easily available and affordable, we hope that this robot would be an active soldier which will sustain our flood supply chain amidst any natural inconvenience.

Keywords:ROS, 6DOF, Joint Angles, Path Finding, Kinematics, Transfer Learning, Deep Learning, Multi-Object Tracking, Deepsort, Data augmentation, Robot Farming, Greenhouse

Dedication

Dedicated to "Bracu Mongol-Tori", where we learnt to dream.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our advisor MD KHALILUR RHAMAN, PhD and co-advisor GOLAM RABIUL ALAM, PhD for his kind support and advice in our work. They helped us whenever we needed help.
And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.
Also honorable mentioning two people and an organization for their endless support.
AQUALINK BANGLADESH Ltd.
SIHAB SAHARIAR, ID : 20101402
S.M.ABRAR MUSTAKIM TAKI, ID : 20301125

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

DOF Degree of freedom

mAP Mean Average Precision

MOT Multiple Object Tracking

ROS Robot Operating System

SORT Simple Online and Real-time Tracking

SOT Single Object Tracking

YOLO You Only Look Onc

Chapter 1

Introduction

1.1 Background

The present human overpopulation tendency, especially the combination of changing consumption habits, growing preference, and organic waste, is putting tremendous strain onto farming infrastructure including ecological assets [17]. As a result, access to nutrition is amongst the most pressing issues facing humanity throughout the late twentieth era [37]. Food is mostly provided through farming systems. Intensively farmed cereals now occupy around 275 million hectares all across the globe [40]. That sector keeps expanding at even a 1.3 percent yearly pace [24]. Although just 23 percent of the agricultural area is used for agricultural grains, they contribute 45 percent of overall livestock farming [36] [32]. The agricultural output should grow over 70 percent before 2050 to meet global consumption [32]. The above-projected rise in global livestock farming necessitates either the expansion of agricultural territory or the augmentation of output on presently cultivated acreage [26]. Regarding moderate cases, a 53 percent incremental use, as well as a 38 percent growth in the cultivated area globally, might be necessary to meet the 2050 agricultural consumption goal [7][39]. The biggest downside of growing farmland is the scarcity that forces farm utilization transformations, resulting in the destruction of natural habitats [41]. As a result of the forest loss systems occurring, transforming terrain into cropland is the second largest worldwide danger to wildlife preservation [20]. The Brundtland Report of the UN World Commission on Environment and Development [41] coined the word "sustainable," as it is currently recognized. In this paper, responsible growth is characterized as meeting specified requirements while jeopardizing later descendants' capacity to fulfill their own [2]. Collectivism, ecological, and financial viability include three elements of socio-ecological systems. It is still seen as a principle, comparable to liberty, fairness, and freedom [2]. Following the publication of this assessment, several significant incidents prompted the creation of accords pushing for Humanity's prolonged stay, such as the Rio Declaration of 1992 [44], which protects the earth's natural biologic mineral wealth and promotes their long-term usage. In addition, the Kyoto Protocol [56] obligates nations involved to limit carbon dioxide pollution, while the United Nations Millennium Development Goals [30] set forth guidelines for improving worldwide living standards and the ecosystem. In order to accomplish this need, it is estimated that the existing farmed area would need to be expanded by 200 million hectares. Throughout the field of organic expansion, the increased land area needed might be worth $10 million. Farm expansion must be based on technology advancements, adaptability, as well as the transmission of extensive farming practices to non-intensive cultivation. In this regard, efficient leadership strategies might significantly cut nitrogen consumption and greenhouse gas emissions. Responsible expansion is an objective. However, there is no such thing as a one-size-fits-all answer. The benefits of many techniques, such as those connected to technological innovation and sustainable farming, must be thoroughly researched and evaluated using macro and microeconomic characteristics [20]. In this approach, increasing productivity on the present farmland is a viable choice, with greenhouse cultivation being among the most significant possibilities.

Food security, as defined by the World Health Organization (WHO), occurs when all people get access to adequate, secure, and healthier meals to fulfill individual nutritional needs and diet quality for such an improved lifestyle at all times [52]. Food security seems to be a four-part paradigm that revolves around the provision of resources, production consistency, entrance to certain sources, and proper ecological utilization of nutrients. As a result, food availability is still a complicated issue. Food security has grown, proliferated, and varied since the 1974 World Food Conference. Traditionally, malnutrition has been concerned with the allocation and accessibility of sufficient food. Nevertheless, quasi issues which have a significant impact on agricultural development, affordability, including accessibility, such as population pressure, wealth disparity, congestion, deterioration of the environment, veterinary care, and external conditions, influence food security toward a large degree. Bangladesh has seen a significant pace of growth in food security, but continued restrictions on access to food and insufficiency for family-level country resources to buy grain, market forces instabilities, and also environmental as personal catastrophes, are all obstacles to agriculture. Bangladesh has been designated as among the nations severely affected by natural disasters. Agro specialists believe that, although increased productivity, Bangladesh's crop yields have to be further diversified. Initiatives towards intensifying as well as diversifying crop yields, as well as increasing its durability, are required to enhance dietary diversity [28].

The idea of food security had already expanded, proliferated, and varied during that 1974 World Food Conference. Traditionally, food security has been associated with the presence as well as accessibility of good nutrition. However, non-food issues that have a significant impact on agricultural supply, usability, and provision, such as haphazard social process , wealth disparity, overexploitation, habitat destruction, veterinary care, and biological influences, impact the quality of life to a large degree. Bangladesh has seen a significant continued robust food production, but continued restrictions on agricultural sustainability including insufficiency of family as well as country resources to buy dietary, market forces instabilities, along with environmental and human occured catastrophes, are all obstacles to agriculture activities. Food security, as per the World Health Organization, occurs even though all individuals possess education and access to adequate, secure, and nutritious food to fulfill individual dietary requirements and diet quality for an adequate standard of living whatsoever periods [18]. Food security is a four-part notion that revolves around the supply of nutrients, supply consistency, entry to certain sources, and the proper ecological utilisation of meals. As a result, nutrition is a complicated issue.

1.2 Research Problem

Environmental uncertainty is posing a severe threat to Bangladeshi farming, which is directly affected by it. Bangladesh is not to high from sea-level and riverine nation that, due to its geological location and underprivileged social and demographic situations, is rated high among nations susceptible to environmental disruption [23]. These are some of the decade's largest problems could be ensuring sustainability of agriculture through consideration of environmental change effects. Bangladesh is among the planet's top calamities nations, and climate-related calamities including severe weather events, dry weather problems, as well as famine likely severely sabotage broader capacity building. Agricultural stability remained seriously challenged in 2007 following multiple extreme weather events and the hurricane Sidr. While Bangladesh is heavily reliant on agriculture, it is still not nutrition to crop production deficits caused by global catastrophes. As a result, numerous natural catastrophes and the resulting reduction in food products increase Bangladesh's severe underdevelopment and malnutrition levels. Bangladesh contains 8.774 million hectares of farmland, of which 88 percent is farmed, leaving just a little amount of room for expansion [10].

According to [23], Bangladesh has been listed as being among the nations most vulnerable to climatic disruption. Agro specialists believe that, notwithstanding increased productivity, Bangladesh's crop yields have to be increased and diversified. Improvements in intensifying and diversifying crop yields, along with increasing its resilience, is required to enhance dietary diversity [28]. Lawmakers should focus many essential areas to adapt again for anticipated problem of feeding the massive inhabitants, including boosting production efficiency, keeping a permanent sufficient inventory of basic living commodities, enhancing agro Investments in research and development, improving domestic and multilateral business to effectively deal with higher input costs, and climate global nutrition disruptions.In recent years, the agriculture industry has grown at a rate of roughly 3% per year on average. Agricultural productivity, on the other hand, is being strained by rising food demand and dwindling resources.

From [16]

Year	1980	1990	2000	2010	2011
Agriculture, value added (% of GDP)	31.55	30.25	25.51	18.59	18.29
Agriculture, value added (annual % growth)	0.16	9.37	7.38	5.24	5.13

Table 1.1: Contribution of Agricultural sector in the economy [16]

Bangladesh is the planet's most populous nation. Fertility projections suggest that by 2050, this figure may have increased to roughly 220 million. Bangladesh is undergoing tremendous demographic expansion as well as on that verge of having catastrophic undernourishment in the nearish term. Undernutrition in Bangladesh is caused by a variety of interconnected variables, the far more notable of which is destitution. The major pressing concerns in regard of dietary accessibility include market growth for an ever-increasing demographic, periodic malnutrition, extreme weather, and ecological pollution. Despite the fact it is an universal challenge, it is now being addressed by considering the reasons of malnutrition at the at the national and regional levels. Malnutrition has quite a number of detrimental effects on the immune system, many of which lead to this problem. Individuals who are undernourished are greater likely to acquire hyperglycemia, heart disease, and various serious medical issues, according to research [9][4][14]. Several issues have financial consequences including on households, however also upon a whole system, since they reduce the engaged working population and lower cognitive and emotional function. Besides that, periodic droughts and excessive living expenses are wreaking havoc on the people in the lower classes' malnutrition, particularly in the nation's northerly regions.

In Bangladesh, malnourishment is responsible for over a quarter of infant fatalities as well as approximately a fifth of pregnancy fatalities each year. As a consequence, Bangladesh has one of the worst rates of starvation within the globe. Malnutrition causes chronic conditions, and as a result, people are unable to support themselves. Prolonged starvation has a number of consequences, including the perpetuation of undernourishment, the reduction of people' capability to do things and deliver a healthy kid, and the deterioration of a baby's ability to think critically and conduct positive and functional lifestyles.

To overcome this issue, innovation of agricultural practices is being explored. Greenhouse farming is one of them. Instruments and controllers that are easy to do and acquire are employed in contemporary environmentally sound technologies. Through a sequence of trials, obtain competitive to improve the efficiency of greenhouses agricultural techniques. Furthermore, digitalization may enable users to explore the farmland solely relying on time-consuming and manpower-intensive traditional activities [22]. Urban agrarian greenhouses are a method of agriculture wherein plants are produced in tiled slab racks to make the most of ground [35], that is a limited commodity in urban, and lateral capacity, that is underutilized in conventional agriculture. Despite the large expenditure, vacant facilities are outfitted into facilities such as shelves and lights, and agriculture begin. Several various duties, including as seed sowing, surveillance, and cultivation, being carried out in a vertical greenhouse, which necessitates a large employment costs, that is both scarce and costly. Physical contact with the events taking place in the elevated horticulture [45] is extremely extremely difficult due to the techniques and processes utilized inside. Farming robots are becoming increasingly popular as an alternative. There has been a lot of study carried out in this field because industrial automation is employed in farming to replace manual input.

Vertical farming is a basic farming method in which plants grow vertically rather than horizontally. Vertical farm design is determined by local resources and facilities, such as towering constructions with several growing beds, either outdoors or inside, on roofs or in a multistory building. Vertical gardens rely only on the support of walls to keep their plants alive.

The major goal of this study is to create an autonomous robot that can navigate to each plant within the vertical greenhouse, detect plants simultaneously to recognize and collect high-quality harvests, evaluate plant development, and identify crop diseases on vertical farming greenhouses. This sort of approach uses less menial effort and saves time throughout the crop cultivation while improving freshness and production. The use of this sort of stability also helps to reduce wasted dietary. We created a robot that has a manipulator with four actuators. The manipulator is made up of three pieces, each with six segments, and it is made up with a claw to increase the manipulator's rigidity [34]. The manipulator's versatile form allows it

to function in restricted spaces while reducing the harm that the manipulator might do to the vegetation and its yield.
On the other hand YOLOv5 is used to detect and track the tomatoes by training on various devices and dataset . Moreover, for classification OpenAi Clip and RestNet 34 is used to classify bulk amounts of leaves data for disease detection. A* search algorithm is used to operate the robot efficiently in the system to move one point to another.

1.3 Research Objectives

1. To develop a human assistance robot.
2. Solving kinematics of our custom 6 DOF manipulator
3. Developing custom ROS packages for the control of robotic system
4. Developing the electronic system of the robot
5. Comparing different deep learning models on different devices
6. Comparing datasets
7. Robust disease detection for plants
8. Harvesting tomatoes
9. Establishing ml processes for tomato detection and tracking

1.4 Scope and Limitation

There is much research work already conducted for object detection. However there is not much work in object tracking.
So, we have a very big scope in object tracking.Our manipulator could be a universal manipulator. We can use different algorithms to traverse through the green house environment in an efficient way.Robot operating frameworks make the system more rigid. kinematics can describe our manipulator and its movement.
There is a lack of greenhouse in Bangladesh for that reason there is lack of testing environment.Robot frameworks can impose an obstacle because of their compatibility because each robot framework develops for specific applications, Though the greenhouse required infrastructure.

1.5 Document Outline

The second section discusses ROS and its components. Kinematics and the UART communication protocol are also discussed. On the other side, there's talk of Data Augmentation, pre-processing, detection, tracking, YOLO variants, and leaf disease categorization. 3rd section. The third section is about applying the concepts. Using the six degrees of freedom manipulator installed on top of the mobile robot, we used a grid array to identify the shortest route to move the robotic system to the intended area, evaluate its surroundings, distinguish things to report, and gather fruits and vegetables. Various algorithms, different deep learning models, robot frameworks, communication protocols, visualization softwares, computer vision, and mathematical computation are all part of the robotic system's software. Finally section four discusses the comparison between various deep learning models in different conditions. Results of shortest path and inverse kinematics application on the six degrees of freedom manipulator.

Chapter 2

Literature Review and Related Works

In this chapter there is discussion on ROS and its materials.It also talks about kinematics and UART communication protocol. On other hand there is discussion about Data Augmentation, pre-processing, detection, tracking, types of YOLO and Leaf disease classification.

2.1 Greenhouse Farming

A greenhouse is a framework building coated in crystal or polymer sheet (opaque and luminous) wherein photosynthesis takes place in a partly or driven primarily atmosphere. Greenhouse technique has played a significant role in bettering land utilization, producing plants in harsh weather, and in places with rainstorms. The greenhouse polymer sheet acts like a sensitive uv filtration. The "greenhouse effect" is a phenomenon wherein ultraviolet irradiance passes into it and traps the heat power released by the things retained within the greenhouse. Advantage es of greenhouse:

1. Improves production as well as purity while shortening harvest time.
2. As a result of the humidity conservation, minimal watering is required.
3. Assists in the development of initial nursery for a variety of plants.
4. Aids in the cultivation of plants under a variety of environmental circumstances.
5. Aids in the production of expensive crops for the trade industry.
6. Hardens freshly grown crops and grafting.
7. Environmental conditions are moderated.
8. It aids in the invasive species management and illnesses.
9. Optimization of microenvironment and pest resistance aids in crop reproduction and the generation of novel seed types.
10. Off-season agricultural production is conceivable.

2.2 SBC

A single-board computer (SBC) is a full microcomputer with microprocessor(s), storage, input/output (I/O), and other capabilities needed for a working system that is constructed on a motherboard. Single-board computers are often utilized as show or research workstations, instructional equipment, or integrated system interfaces. Several kinds of personal computers and other digital workstations use an one pcb that houses all of its functionality.
According to [48],with its constant evolution from the time of inception towards the present, it has almost no bounds. Although machines are huge enough to occupy a whole house since they first start up, they provide chances to accomplish a variety of tasks by expanding their throughput and shrinking to fit in the palm of one's hand. Compact, transportable, minimal, heavy processors with surprisingly high functionality are sought in particular applications, rather than costly and substantially large systems. Single-board computers are preferable in these situations. Single-board systems contain microchips, storage, input/output, and other key competencies and are constructed on a single integrated circuit [36]. Single-board systems with broad application and cheap cost have been more popular in recent years, owing to their potential to integrate with other sectors of innovation [37]. Single-board computers' technical advancements, as well as its orientation to personal computer performance, have bolstered their application in automation.

2.2.1 Raspberry Pi

The Raspberry Pi is a tiny, low-power microprocessor around the size of a credit or debit card that links to a display unit or television through a standard interface. It's a handy little device that enables people of any age to understand on computing and computer languages like R and Python. It can do all of the functions of a computer, including browsing online and streaming high-definition video, as well as spreadsheet, Microsoft Word, and gaming.
The Raspberry Pi has the capacity to converse with its surroundings and is used in a variety of online creative applications, including audio generators and parental monitors, as well as environmental sensors and posting on social media birdhouse with thermal imaging according to [35]. We would like to witness the Raspberry Pi utilized by children all across the globe to discover how machines function.
Since then, various generations and variants of the Raspberry Pi were introduced. The first Raspberry Pi had a single-core 700MHz processor but only 256MB of RAM, but perhaps the most current edition has a quad-core 1.5GHz processor and 4GB of RAM. The Raspberry Pi has usually cost around $35 USD, with the Pi Zero is one of the most affordable at around $5.
People use the Raspberry Pi all across the globe to study coding, develop electronic initiatives, automate their homes, deploy Kubernetes systems and Cloud technologies, and even utilize it in commercial processes.
The Raspberry Pi is a low-cost computer that runs Linux and contains a set of GPIO connectors for manipulating system components and exploring with the Internet of Things (IoT).

2.2.2 Nvidia Jetson Nano

The Jetson Nano is a compact, intelligent device for hardware implementations and Artificial intelligence - based IoT that packs contemporary AI capabilities into a $99 module. Jetson Nano provides the speed and functionality required to handle current AI operations, allowing you to incorporate sophisticated AI into your latest application quickly and easily. The NVIDIA Jetson Nano is an integrated framework (SoM) and development kit that has a 128-core Maxwell GPU, quad-core ARM A57 64-bit CPU, 4GB LPDDR4 memory, and support for MIPI CSI-2 and PCIe Gen2 high-speed I/O.
According to [39], the simplest option is to utilize your Nano as a rudimentary computer. The majority of other SBCs either do n't possess a GUI and can only run pre-flashed programs, or have a very rudimentary, low-resolution GUI. With the exception of the kernel being written for the ARM CPU and a few other modules Nvidia determined were vital to include on their setup image, Nano runs pretty much complete Ubuntu Linux.

2.3 Micro-Controller

A microcontroller (usually known as a microcontroller unit or MCU) is a related to internal circuit that is normally had for a specific procedure and intended to carry out specific tasks. Utilities, electrical devices, automotive powertrain control equipment, and workstations are all products and gadgets which must be continuously regulated in particular conditions, but microcontrollers may be used for much more.

A microcontroller, in essence, receives data, evaluates it, and then executes a certain function based on the information gathered. Microcontrollers normally run at modest speeds, between 1 and 200 MHz, and must be designed to use less energy since they are incorporated into other electronics that may require more energy elsewhere.

2.3.1 Arduino

Arduino is a completely free foundation for building electronic projects. Arduino consists of a hardware customizable pcb device (also known as a microcontroller) and an IDE (Integrated Development Environment) that runs in the background and is used to write and publish code to the arduino hardware.
The Arduino framework has been more popular among individuals that are just starting up with circuits, and for good reason. Unlike many other previously programmable microchips, the Uno does not require a separate hardware (known as a programmer) to upload new code; instead, all that is necessary is a USB connection. Moreover, the Arduino IDE simplifies programming by employing a basic C++ method. Finally, Arduino has a streamlined design that isolates the functions of its microcontroller into a more digestible package.

2.4 Sharp Distance Sensor GP2Y0A

A PSD (position sensitive detector), an IRED (infrared emitting diode), and a frequency synthesis circuit make up the GP2Y0A ultrasonic depth sensor. Because the triangulation technique is used, the object's brilliance, the ambient warmth, and the operation time are not readily changed by the depth detection. The voltage proportional to the detecting range is produced through this gadget. As a result, these precise distance sensors are a popular option for many applications requiring precise readings. This detector is less expensive than ultrasonic optical viewfinders, but it performs far superior than other IR options. The solitary analogue outlet may be linked to a transceiver inverter for coordinates, or the outcome can be linked to a benchmark for rules have been developed, which is how often these microcontrollers are interfaced. This edition's field of view is around 10 cm to 80 cm (4 to 32).

2.5 A* Searching Algorithm

According to, [58] A * algorithm is a searching method that looks for the shortest route between two states. It's utilized in a variety of applications, including maps.The A* method is used in maps to find the lowest distance between a start state and a goal state.
A* algorithm generally takes 3 parameters to calculate the shortest path to reach the goal:

1. g: the cost of getting from the first cell to the present one. It's basically the total of all the nodes visited since the first.
2. h: It is the predicted cost of travelling from the current cell to the end cell, which is called the heuristic value. The last cell must be reached before the exact cost can be determined. As a result, h represents the expected cost. We must make certain that the cost is never underestimated.
3. f: it is the addition of g and h. f = g+h

2.6 Dijkstra Algorithm

From [54], While guess-and-check may typically determine the shortest route on a small network, the focus of this section is to create ways for solving big issues in a systematic approach using algorithms. A step-by-step technique for addressing a problem is known as an algorithm. The shortest route between two vertices is found using Dijkstra's method. Dijkstra algorithm's working principle:

- Make a zero-distance mark on the last vertex. Make this vertex the current one.
- All vertices that lead to the present vertex should be found. Calculate the distances between them and the finish line. We just need to add the most recent edge since we already know how far the present vertex is from the finish. If this distance exceeds a previously reported distance, don't record it.
- Make a note that the present vertex has been visited. This vertex is not going to be looked again.
- Return to step 2 and designate the vertex with the shortest distance as current.

2.7 PID

Proportional Integral Derivative is the abbreviation for Proportional - integral - derivative Derivative. It's a type of gadget being used industrial applications to control pressure, flow, temperature, and speed among other process factors. To regulate everything, a closed loop system feedback device is used. In this controller, there are process variables. This kind of regulator is used to control the direction of a system when this is otherwise level, toward a target point It's used in almost every situation. temperature control, scientific methods, automation, and a variety of other applications a diverse set of chemical uses The controller makes advantage of closed-loop feedback. to maintain the genuine output of a procedure as close to the aim as possible if possible, else output near the apex of the fixed.

P-proportional: A proportional or P-output controller's is proportional to its input. The is the current mistake (t). The goal or ground pin is contrasted to the true or current situation. It is vital to get feedback. Multiplying the resulting yields the output. A proportional constant is used to calculate the error. If the error value is 0, its control signal is 0. This controller requires prejudicing or manual reset when used alone. This is because of given notion that it never reaches steady-state It maintains the steady-state condition. While ensuring steady functioning, there was an error. Once the exponent Kc is increased, the rate constant Kc. The response time increases.

I-integral: Because of p-controller has the disadvantage of always having an output. An I-controller is necessary to eliminate the mismatch between both the variables and the setpoint. The mistake that is stable It gradually integrates the mistake till the amount of the error is determined. closes in on zero When an error occurs, it saves the state of last control device zero. Integral control decreases its output when a negative error occurs. It slows down. The system's response time is slowed, and its stability is jeopardized. The rapidity with which you respond. Reducing the integral gain, Ki, improves the performance.

D-derivative: The I-controller is incapable of foreseeing improper behavior. in the near future As a consequence, it reacts appropriately whenever the setpoint is changed. D-controller overcomes this problem by anticipating how the mistake will behave in the future. Its results is calculated by multiplying the rate of increase in mistake over time by the derivative. constant. It improves system responsiveness by giving the output a head start. The D controller is more responsive than the PI controller in the figure above. The output settles faster thanks to the controller. It improves the system's overall stability by taking into consideration the latency of the I-phase controller The response time is sped up by a factor of two. boosting the gain on derivatives Finally, we realized that by combining three different techniques, we were able to get the best results. We can get the required system response by using controls. PID algorithms are created with this in mind. various manufacturers in different ways

2.8 Data Augmentation

Data augmentation is a term used in data analysis to describe methods for enhancing the quantity of data available by adding slightly modified copies of existing data or developing new artificial data from previous data. It acts as a regularizer while developing a machine learning model, reducing overfitting. Image augmentation entails changing the training images to create a synthetic dataset that is larger than your original dataset, with the goal of improving your model's downstream performance. Flipping, rotating, cropping, shear, exposure, saturation, adding noise, jittering, and other data augmentations are just a few examples.

1. Flipping: If we reverse the entire rows and columns of an image pixels in vertically, then it is called vertical flip.Horizontal flip occurs when all of the columns and rows are flipped horizontally.
2. Rotating: A source image is randomly rotated clockwise or counterclockwise by some degrees. Which changes the position of the object in frame.
3. Cropping: Random cropping creates a random subset of an original image.
4. Shear: An image will be distorted along an axis.
5. Exposure: Exposure finds out the amount of black or white which is added to the colors. If the value goes higher, that will be greater in variance.
6. Saturation: It adjusts how vibrant the image is.
7. Adding noise: Expands the size of the training dataset.
8. Jittering: Randomly change the brightness, contrast and saturation of an image. [51]

2.9 Data Pre processing

Accordindg to Techopedia [52],Data preparation is a vital stage in Machine Learning that enhances data quality and makes it easier to extract useful knowledge from big data. Data preprocessing is a term used in Machine Learning to explain the method of sanitizing basic data to make it suitable for the creation and training of Machine Learning models. Data preprocessing, in basic words, is a data mining strategy being used Machine Learning that converts raw data into a readable and understandable format. There are numerous methods for pre-processing:

2.9.1 Auto-orient:

Auto-orient removes the EXIF data from your photographs, allowing you to see images in the same orientation as they are saved on disk.

2.9.2 Resize:

By resizing, the size of the image can be changed to desired dimensions. Annotations are scaled proportionally.

2.9.3 Grayscale:

Converts a multi-channel RGB picture to an one grayscale channel. This conserves memory. The weighted total of the matching red, green, and blue pixels may be used to derive the value of each grayscale pixel: $Y = 0.2125\ R + 0.7154\ G + 0.0721\ B$.

2.9.4 Auto-Adjust Contrast:

Enhance images which are low in contrast. There are some variants of adjusting the contrast of images.

- Contrast Stretching:All intensities between the second and final and 98th percentiles are included in the rescaled picture.
- Histogram Equalization: In a picture, "spreads out the most intensity values."
- Adaptive Equalization: Adaptive Histogram Equalization with a Low Contrast Limit (CLAHE). A local contrast enhancement approach that use histograms generated over multiple tile sections of the picture.

2.9.5 Modify Classes:

When creating a new edition of your dataset, you may use a preprocessing tool to remove certain categories or remap (rename) classes.

2.9.6 Tiling:

When it comes to recognizing little things (particularly in scenarios like aerial imaging and microscopy), tiling can be useful. [52]

2.10 Object Tracking

Object tracking is a deep learning application in which the program takes a series of initial object detections and creates a unique identifier for each of them, then tracks the detected objects as they move around frames in a movie. In other words, object tracking is the task of automatically identifying objects in a video and interpreting them as a set of trajectories with high accuracy.

Multiple Object Tracking Algorithm Stages

Stage 1: Designation or detection: Objects of attention are recorded and highlighted during the designation phase. The algorithm examines input frames in order to identify items that correspond to goal classes. Bounding boxes are utilized to do detection as part of the approach.
Stage 2: Motion: Detection is examined by feature extraction algorithms in order to obtain characteristics such as look and interaction. In most cases, a movement predictor is used to anticipate the future location of each monitored target.
Stage 3: Recall: Using feature predictions, similarity scores among detection couplets are obtained. The scores are then used to connect detections that are linked

to the same target. Similar detections have the same ID, whereas and those who aren't apart of a pair have a different ID.

There are some object tracking algorithms

2.10.1 OpenCV Object Tracking:

OpenCV is a popular option since it offers several algorithms that are especially tuned for the objectives and goals of item or movement tracking. Among the OpenCV object trackers are the BOOSTING, MIL, KCF, CSRT, MedianFlow, TLD, MOSSE, and GOTURN object trackers. Each of these trackers is better suited to a certain objective.

2.10.2 DeepSORT:

DeepSORT is a well-known object tracking framework that also happens to be one of the most often utilized object tracking methods. By including aesthetic information into the algorithm, DeepSORT's performance is significantly enhanced. Because the integration, objects may be tracked over longer durations of occlusion, reducing the incidence of identity changes. For its simplicity, it is the quickest of the bunch, averaging 16 frames per second while retaining acceptable accuracy.

2.10.3 Object Tracking MATLAB:

MATLAB is a numerical computing platform that differs in implementation from DeepSORT and OpenCV, yet it is still a viable option for visual tracking tasks. The Machine Vision Toolbox in MATLAB includes video tracking algorithms including such update mean shifting (CAMShift) and Kanade-Lucas-Tomasi for monitoring a single element or as building blocks in a more elaborate tracking system (KLT).

2.10.4 MDNet:

The R-CNN object recognition network inspired MDNet, a Convolution neural visual tracking service. It's swift and precise. It operates by running samples of potential candidate sites through a CNN. The CNN is typically trained on a huge data and perfectly acceptable at first frame of such an input video. As a result, MDNet is best suited to real-time object tracking scenarios. It is a dependable solution despite its high computing complexity of speed and space.

2.10.5 SiamMask:

Using fully convolutional siamese networks, a simple multi-task learning technique is developed. SiamMask runs online, creating class agnostic object recognition mask and rotational cluster centers at 55 frames per second after being trained with a single boundary box initialization.

2.10.6 TrackRCNN, Tracktor++ and JDE:

TrackRCNN provides segmentation but hard to use in real life tracking (1.6 FPS). Tracktor++ is accurate but not viable for real time tracking. JDE is great for accuracy and frame rate but in low resolution.

2.11 Transfer Learning

Transfer learning is the application of a previously learned model to a new problem. It has gained a lot of traction in the Deep Learning community because it enables you to build Deep Recurrent neural Network with very little input. This is extremely useful because most real-world problems lack the thousands of labeled pieces of data required to train such complex models. Through transfer learning, we want to apply what we've learnt in one activities to improve generalization in another. The values that a Network acquires in Task A are transferred to a new Task B.

2.12 YOLO

YOLO is a neural network-based real-time object recognition technology. **section'**. This algorithm is particularly popular for its speed and accuracy. In a number of applications, it has been used to recognize traffic lights, people, parking meters, and animals.
YOLOv5 is available in four models, namely s, m, l, and x, each one of them offering different detection accuracy and performance as shown below

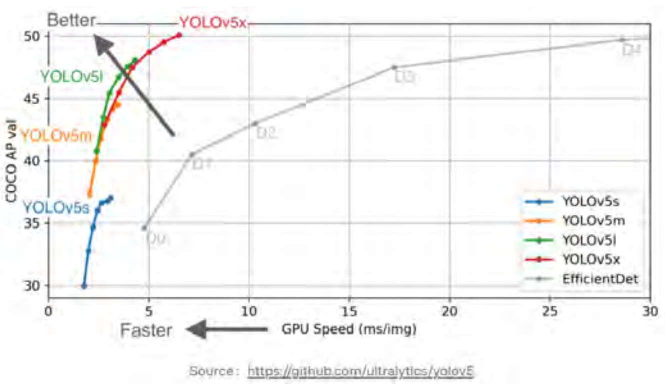

Figure 2.1: Detection Accuracy of different YOLO algorithm

YOLO algorithm works by following three techniques:

Model	size	AP^{val}	AP^{test}	AP_{50}	$Speed_{V100}$	FPS_{V100}	params	GFLOPS
YOLOv5s	640	36.8	36.8	55.6	**2.2ms**	**455**	7.3M	17.0
YOLOv5m	640	44.5	44.5	63.1	2.9ms	345	21.4M	51.3
YOLOv5l	640	48.1	48.1	66.4	3.8ms	264	47.0M	115.4
YOLOv5x	640	**50.1**	**50.1**	**68.7**	6.0ms	167	87.7M	218.8
YOLOv5x + TTA	832	**51.9**	**51.9**	**69.6**	24.9ms	40	87.7M	1005.3

Source: https://github.com/ultralytics/yolov5

Figure 2.2: Detection Accuracy of different YOLO algorithm

1. Residual block: The image is initially divided into many grids. Each grid has the size S x S. The illustration below shows how a square is generated from a source images.
2. Bounding box regression:A frame is an edge in an image that attracts attention to a specific item. Every frame in the picture has the qualities listed below:
   (a) Width
   (b) Height
   (c) Class
   (d) Bounding box center
3. Intersection over union (IOU):In object detection, the intersect over union (IOU) notion depicts how boxes overlap. YOLO employs IOU to construct an exporting frame that appropriately encompasses the objects.
   Each grid cell predicts the bounding box coordinates and their reliability ratings. The IOU is 1 if the expected and actual boundary boxes are identical. Barreling boxes who are not the identical size as the real box are removed using this method.

2.13 Model Evaluation Metrics

Evaluation metrics are used to judge the efficacy of a data - driven machine learning model. Any endeavor will necessitate a review of machine learning techniques or algorithms. A number of assessment measures may be employed to put a model to the test. Among them are accuracy of the classification, proportional loss, classification error, and other measures.
Classification Accuracy: Ratio of the number of correct predictions to the total number of input samples

Logarithmic Loss: works by penalizing the false classifications
Confusion Matrix: A matrix which gives the complete performance of a model

2.14 ResNet-34

ResNet-34 is a classification model for images. Picture classification is an important problem where a model is trained using labeled sample images to recognize a set of target classes (items to identify in pictures). Early image processing algorithms used raw pixel data as their input. Resnet34 is a 34-layer deep neural networks image classification model that is state-of-the-art. The Matching dataset, which contains 100,000+ photographs organized into 200 categories, was used to pre-train this model.

2.15 ROS

Developing firmware for robots is challenging, especially as the size and breadth of automation expands. As different kinds of robots possess vastly differing technology, programming for them becomes challenging. Furthermore, the sheer quantity of the necessary code, which must include a deep stack beginning with driver-level software and extending up to perception, abstract thinking, and beyond, might be intimidating. Software architectures of robotics must also allow large-scale software integration activities, since the needed breadth of knowledge is considerably beyond the capability of any one researcher [13].

To address these issues, several robotics academics have developed a number of frameworks to manage the complexity and promote quick software development for experiments, culminating in several robotic software systems presently in use in academia and industry [8]. There are several robot software frameworks such as CARMEN, RDS, ROS, MOOS, Player/Stage etc [19].

The Robot Operating System (ROS) is a free and open source robot software platform. ROS is not an operating system in the conventional sense of process management and scheduling, but rather a structured communications layer that sits on top of the host operating system [13]. ROS is like a traditional computer operating system like Linux or Windows which offers a variety of programs that a user can use. With ROS, the user can use a collection of tools and programs to control and simulate a system of robots [19]. Many difficulties that humans consider simple might really have a wide range of differences across instances of tasks and settings from the robot's viewpoint [31]. Consider bringing a cup of coffee from the kitchen to your desk. For a normal human, this is an easy task but for a robot to navigate, recognize, and use it's mechanical structure is very difficult. Frameworks such as Robot Operating System make the process structured and efficient.

The framework Robot Operating System has several parts. It lets the user read data from sensors, control actuators and motors. To control the hardware, ROS has vast and increasing sets of algorithms for robots that allow the robot to create maps of the world around it, move through the map while performing a variety of tasks. Developers can visualize the different states, actions and behaviors of a robot. Furthermore, ROS has a large ecosystem that includes documents about many aspects of the system and community support. The development of ROS follows the philosophy of Unix software development [31].

Although the Robot Operating System is open source, it is maintained by Open Robotics. Since it's first release, the original ROS has had several versions and recently, ROS 2 was released. Since ROS is open source, tools for new releases are developed fairly slowly. Furthermore, in many cases ROS 1 out performs ROS 2 but ROS 2 has more features and improvements .

The framework has a large array of individual programs that communicate with each other. The individual programs follow some key concepts of ROS shown in figure 2.3.

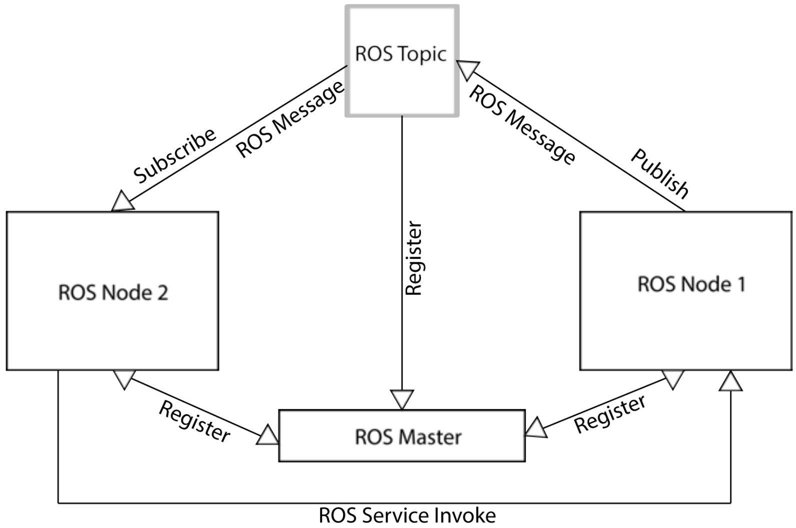

Figure 2.3: Key concepts of ROS

2.15.1 ROS Master

The ROS Master is responsible for providing name as well as identification functions to the ROS system's nodes. It keeps track of topics, services, publishers and subscribers. The Master's job is to make it possible for different ROS nodes to talk with one another. After nodes connect to each other, they use peer to peer communication. The Parameter Server is likewise provided by the Master [31].

2.15.2 ROS Nodes

ROS nodes are executable programs that do a specific task. Afew nodes doing the same kind of tasks will form a package. Nodes communicate with each other using ROS topics, ROS services and ROS messages. When a node is executed, it contracts ROS Master to register itself in the network. A robot using the framework will have many nodes. For example, one node receives data from a sensor, another node controls the wheels. Moreover, nodes can be of 2 types, publisher nodes and subscriber nodes [31]. Publisher nodes transmit specific messages via specific topics to subscriber nodes that receive the messages from the topics using a callback function. A node can be a publisher or a subscriber or both. Also, the communication between nodes can be 1:N, N:1, N.

2.15.3 ROS Topic

A topic is the designation given to a stream of communications of a specific message [13]. The data that a sensor gives needs to be published, the publisher node transmits the data over a topic, for example "/sensor_data". This topic system makes ROS appealing to developers as setting up ways to transmit data can get very messy. Data types of the messages on a topic should be the same.

2.15.4 ROS Message

ROS messages are a common method to convey data across topics in ROS. Although there are pre-installed message formats, you must adhere to ROS standards when creating your own messages [31]. For example, if you wish to transmit 4.00, you must use Float64 since it is a float type number. Furthermore, The transmission of ROS messages is performed at a constant pace.

2.15.5 ROS Service

The publish/subscribe communication mechanism is very adaptable. It is transmitted continually at a set pace, as we saw with ROS messages. However, certain messages could only have to be sent occasionally. For example, switch on a fan or a lamp and then turn it off. ROS Service has been used to do this. The publisher will request that the subscriber switch on a machinery, and the subscriber responds with "done."
In order to assist developers and researchers, ROS has large sets of tools.

2.15.6 ROS Graph

The ROS Nodes create a network between themselves. This network is known as computation graph [38]. Computation graph contains ROS messages, Services, Master, Nodes. As we know that robots with ROS can contain several ROS nodes, topics, messages etc. ROS Graph keeps track of all ROS processes.

2.15.7 TCP ROS

TCPROS is used by ROS Messages and Services as the transport layer. It transports message data using conventional TCP/IP sockets [59]. Inbound connections are received via a TCP Server Socket with a header providing information about the packet data type and forwarding.

2.15.8 Gazebo

Gazebo [55] is a simulator that can be used to simulate complicated interior and exterior robotic scenarios[38]. Sophisticated robots, robot sensor systems, and a range of 3D objects can all be simulated. In its library, Gazebo currently offers simulation models of renowned robots, devices, and a range of 3D objects. We don't need to build anything in order to utilize these models. Gazebo provides a nice interface for ROS which provides all of Gazebos controls. We could install Gazebo without ROS, however the ROS-Gazebo interface is required to interact between ROS and Gazebo.

2.15.9 RViz

RViz is a toolbox for visualizing data created by configurable data modeling in the real world. The RViztookit is simple to use and is unaffected by data structures or methods. The architecture of the RViz toolkit is built in such a manner that toolkit developers may implement just a limited number of functions, and each function is basic and modular, allowing the toolkit to be readily transferred to many systems [29]. As a consequence, we can quickly view analytics and data structures, which aids engineers in debugging and profiling compiled code efficiency. To implement RViz and ROS together, ROS-RViz interface is needed.

2.15.10 URDF

A robot having links interconnected by joints in a network or tree may indeed be modelled using URDF. Networks of joints offset by links may be used to simulate most Commercial Robots. A tree data architecture with joints linked by connectors may be used to represent robots [33]. To make a robot's linkages and joints, there are two kinds of URDF XML components. Inertial characteristics, visual assets, and collision attributes may all be found in link elements. Origin, primary link name, child link name, direction of pivot, calibration, kinetics, limit, imitate another joint, and safety controller information may all be found in joint elements.

2.16 Kinematics

Kinematics is concerned with the motion of bodies in a robotic device regardless of the forces or torques that generate the motion. Kinematics is perhaps the most basic part of robot design, evaluation, control, and simulation since robotic systems are by their very nature built for motion. To tackle core kinematics difficulties, the robotics field has concentrated on efficiently applying various representations of position and orientation, as well as their derivatives with regard to time [34]. Robotic systems are structures of rigid bodies linked by joints. The pose is the collective name for the posture and direction of a structural body. Robot kinematics, then, is the study of the posture, velocity, acceleration, and any higher-order variations of the posture of the bodies that make up a mechanism. System of bodies can be connected in various ways. Kinematics describe the connection of a body to another, topologies. Serial chains, parallel mechanisms and tree structure. A serial chain forms when solid bodies are linked to each other and each body is connected to two other bodies. Additionally, the first and the last body in the serial chain only has one other connection. Serial chains are a variation of the tree structure. Because a tree is built by solid bodies linking to each other, forming branches and each branch may have multiple other solid bodies connected to it. Serial chains are created when there are no branches, just one connection on each side. Furthermore, a parallel manipulator kinematic chain of solid bodies with numerous separate kinematic chains connecting the end-effector to the base. This forms a closed loop in the kinematic structure. The end-effector of parallel robots is generally linked to two or more chains that are connected to at least one actuator.

Given a chain of solid bodies, the last solid body's coordinates can be calculated using kinematics. Depending on the application, two types of kinematics can be used. But to understand the two types of kinematics, some key concepts need to be understood.

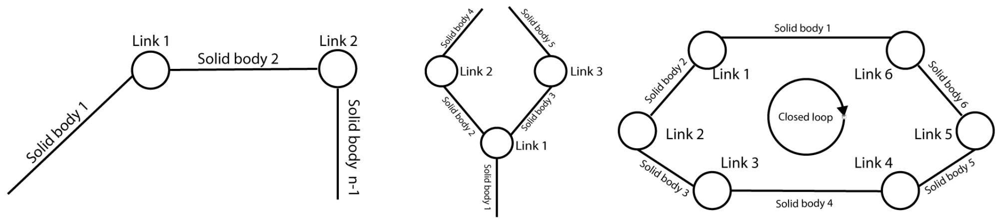

Figure 2.4: Serial Chain, Tree Structure, Parallel Mechanism

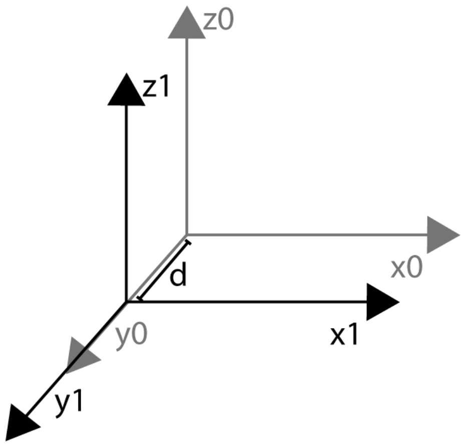

Figure 2.5: Displacement

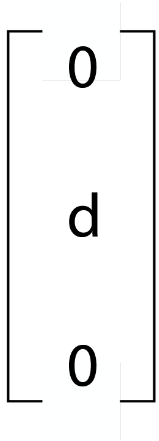

Figure 2.6: Displacement Vector 1

2.16.1 Displacement

Displacement refers to the movement of a kinematic structure from one position to another. The 3x1 vector represents the location of the beginning of reference frame i compared to reference frame j [6]. The 3x1 vector represented in matrix (shown in figure 2.5) form describes the change in position along the x, y and z direction [43].

$$ {}^j p_i = \begin{pmatrix} {}^j p_i^x \\ {}^j p_i^y \\ {}^j p_i^z \end{pmatrix} $$

This vector's components are the Cartesian coordinates of p_i inside the j frame, which are the vector ${}^j p_i$'s projections onto the respective axes. The vector components may alternatively be represented in terms of p_i's spherical or cylindrical coordinates in the j frame. Such representations are useful for analyzing robotic devices with circular and cylindrical joints. Displacement is only concerned with the change of a solid body's position in a kinematic chain. It does not consider the rotation vector. Additionally, When the rotation of a solid body changes, the origin of the body is unchanged. As a result, the displacement vector remains in the same position.

2.16.2 Rotation

A rotation is a dislocation in which at least one position in a rigid body retains its original location but not all lines in the body stay parallel to their original orientations. An object in a circular orbit, for instance, rotates along an axis running through the center of its path, and every position on the rotational axis represents a point in the body that stays in its original location [3]. Any representation of ori-

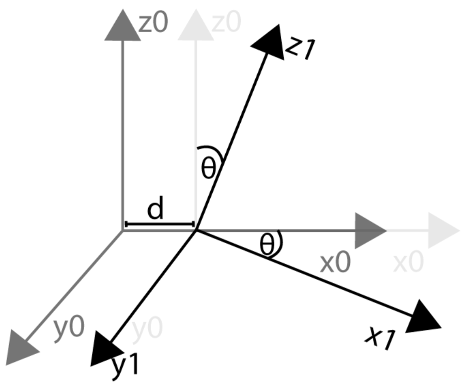

Figure 2.7: Displacement and Rotation

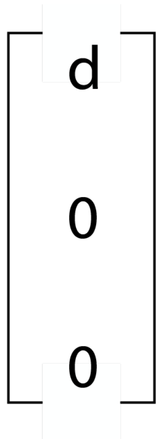

Figure 2.8: Displacement Vector 2

entation, like any representation of position or translation, may be used to produce a description of spin, and vice versa.
The basis vectors m_i n_i o_i may be expressed in terms of the basis vectors m_j n_j o_j to describe the orientation of coordinate frame i (change in latex) relative to coordinate frame j (change in latex). This produces $j_{\widehat{m}_i}$ $j_{\widehat{n}_i}$ $j_{\widehat{o}_i}$, which is defined as the rotation matrix when represented as a 3x3 matrix. The dot products of the basis vectors of the two coordinate frames make up the components of j_{R_i}

$$j_{R_i} = \begin{pmatrix} \widehat{x}_i \cdot \widehat{x}_j & \widehat{y}_i \cdot \widehat{x}_j & \widehat{z}_i \cdot \widehat{x}_j \\ \widehat{x}_i \cdot \widehat{y}_j & \widehat{y}_i \cdot \widehat{y}_j & \widehat{z}_i \cdot \widehat{y}_j \\ \widehat{x}_i \cdot \widehat{z}_j & \widehat{y}_i \cdot \widehat{z}_j & \widehat{z}_i \cdot \widehat{z}_j \end{pmatrix} \tag{2.1}$$

Dot products of two vectors produce cosine of the angles involved [21]. Rotation matrix contains dot products. As a result, a rotation of θ degrees on the j_{m_i},j_{n_i} ,j_{o_i} axis of i frame is

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) \\ 0 & sin(\theta) & cos(\theta) \end{pmatrix} \tag{2.2}$$

$$R_y(\theta) = \begin{pmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{pmatrix} \tag{2.3}$$

$$R_z(\theta) = \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.4}$$

When a chain of solid bodies change positions, their rotation matrix can differ based on its movement. Finding the rotation matrix involves multiplying the rotation matrix when there is only displacement and no rotation with the rotation matrix when there is both displacement and rotation. Combining Figure 2.5 and Figure 2.7, the body will have displacement and rotation. Rotation matrix of figure 2.5 is

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation matrix of figure 2.7 equation (2.1). So multiplying Figure with identity matrix and equation (2.1) give us

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) \\ 0 & sin(\theta) & cos(\theta) \end{pmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) \\ 0 & sin(\theta) & cos(\theta) \end{pmatrix}$$

2.16.3 Kinematic diagram

When describing the connectedness of links and joints using geometric objects, understanding of kinematic diagram terminology and representational format is essential [46]. The joints enable various sorts of movement, such as rotation and translation, while the links are structural features, such as a handle or a screw. As a result, a kinematic diagram is a standardized graphical representation of a mechanism's linkages and joints.

Polysemic as well as monosemic systems are used to describe the two forms of informal and formal systems, respectively. Multiple interpretations of each visual element, like the grey rectangle or the jagged white form in the figurative artwork, are possible with polysemic systems. In formal rule-based representations, on the other hand, the standard determines how a graphical element, such as a rectangle in a kinematic graph, is assigned meaning.

2.16.4 Six degree of freedom

According to [25], 6 degrees of freedom is the minimum needed to reach a volume of space from every angle. In a human arm there are at least six degrees of freedom. If your robot arm can do all of these motions then your arm can do pretty much any job a human can.

Axis 1: Let your arm hang down. Raise it in front of you, turning only at the shoulder.

Axis 2: Let your arm hang down. Raise it out away from your body like you're jumping jacks.

Axis 3: Bend your elbow.

Axis 4: Twist between the wrist and the elbow.

Axis 5: Make a fist, then make it nod up and down. Bonus points if you draw eyes on the first knuckle of your index finger.

Axis 6: Envision holding an enormous dial in your fingers. Turn the dial without moving the rest of your hand.

2.13.4.1 Forward Kinematics

From the base frame to the end-effector, a manipulator is made up of serial links that are connected by revolute or prismatic joints. Forward kinematics is the process of calculating the location and orientation of an end-effector in relation to joint variables. A proper kinematics model should be used to provide forward kinematics for such a robotic system in a systematic way. The most frequent way for expressing robot kinematics is the Denavit-Hartenberg method, which has four parameters [5]. The connection length, link spin, link offset, and joint angle are represented by the parameters. To calculate DH parameters, each joint has a coordinate frame associated with it. The coordinate frame's Zi axis points in the direction of the joints' rotating or sliding motion. Considering the geometric shape of the robotic system and the values of a set of joint locations equal to the number of data points of the mechanism, a more general definition of the forward kinematics issue is to identify the relative location and direction of any two specified members [34]. Because joint locations are often recorded by sensors positioned on the joints, and it is important to determine the locations of the joint planes relative to the reference frame, the forward kinematics issue is crucial for creating robot coordination algorithms.

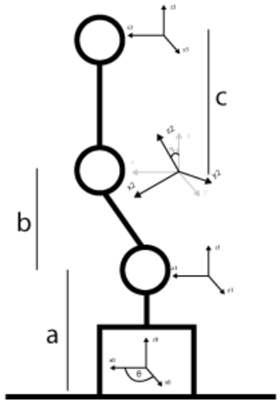

Figure 2.9: Coordinates in a manipulator

Figure 2.9 shows that the distance between z_1 and z_0 observed across z_0 is given as a, the degree between z_1 and z_0 recorded along x_i is assigned as i_{-1}, the distance between x_1 and x_0 calculated across z_i is written as d_i, and the angle between x_1 and x_0 recorded about z_1 is allocated as θ [12]. For a single connection, the generic transformation matrix ${}^{i-1}_{i}T$ may be calculated as follows

$$
{}^{i-1}_{i}T = R_x(\alpha_{i-1})D_x(a_{i-1})R_z(\theta_i)Q_i(d_i)
$$

$$
\begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}
$$

$$
{}^{base}_{end_effector}T =^0_1 T^0_1T \cdots^{n-1}_{n} T \tag{2.6}
$$

$$
{}^{base}_{end_effector}T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.7}
$$

$$
{}^0_6T =^0_1 T(q_1){}^2_3T(q_2){}^3_4T(q_3){}^4_5T(q_4){}^4_5T(q_4){}^5_6T(q_6) \tag{2.8}
$$

2.13.4.2 Inverse Kinematics

Forward Kinematics' inverse function/algorithm is Inverse Kinematics [1]. The Inverse Kinematics component takes a target location as input and generates the posture necessary for the end effector to achieve the target position [11]. Inverse Kinematics takes care of all the difficult computational effort involved in determining the posture. This is shown in Figure 2.10. There is a serial chain of solid bodies in A1 position and direction. It establishes a target location for the end effector to attempt to achieve. After applying inverse kinematics, the bodies move to A2 position and direction.

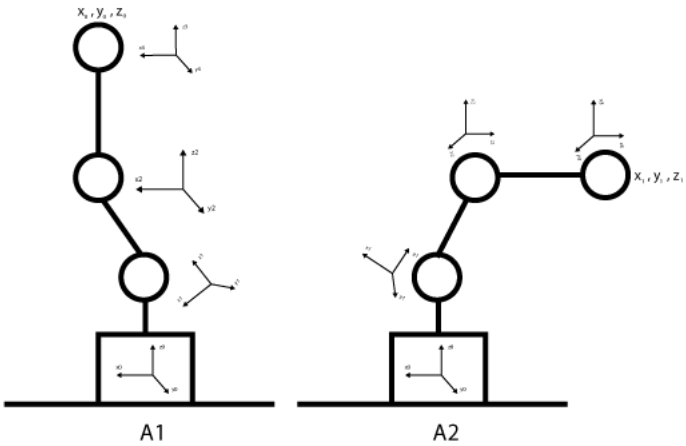

Figure 2.10: Inverse kinematics applied to a serial chain of solid bodies

Solving inverse kinematics to find the angles of the serial chain solid bodies has several ways but two of them stand out in performance and real world conditions [11].
One of the approaches is the analytical approach. This process involves an extensive usage of trigonometry. One of the analytical approaches starts off with assumptions about the angles to be calculated. Firstly, the kinematic diagram of the serial chain is drawn which is crucial for finding the desired angles of rotation. Because, the inverse kinematics equations are obtained from the kinematic diagram. If the serial chain has 6 or more degrees of freedom, the location of the end-effector is mainly calculated by finding the first three angles of the serial chain. The remaining joints are used to orientate the end-effector. The analytical approach uses forward kinematics to obtain the rotation matrix which uses the base coordinate axis as the reference frame. The rotation matrix is then inverted. Calculation on the last joints need to be done as well. Again, the rotation matrix is calculated by applying forward kinematics. The matrix is then inverted. The both matrices are then combined to get the rotation matrix of the whole serial chain. Afterwards, the rotation matrix is applied to the inverse kinematics equations to determine serial chain link angles to reach the desired x, y, z coordinates.
The numerical inverse kinematics approach solves the problem by iterative calculations of the algorithm to find the joint angles. Numerical approach has several methods, but Pseudo Inverse Jacobian Method is mostly accepted by academia. The kinematic diagram is also needed in the numerical approach. Rotation and displacement matrices are calculated using the kinematic diagram. Rotation matrix can be

obtained using either Euler angles, Quaternions and Axis-Angle representation [1]. Axis Angle representation describes a robotic mechanism in the 3D axis in a fixed orientation and angle. The rotation and displacement matrix are multiplied and transformed to build a homogeneous representation. After that, the position of the serial chain is determined by the homogenous representation calculated in the global frame. The position of the serial chain and the last solid body is used to find the error between the desired position and the current position. This calculation is done iteratively, making small changes towards the desired position. A matrix is used to store the changes. Small interactions are used because calculating the displacement is simpler and the Jacobian matrix can be applied. The Jacobian matrix is as follows

$$J = \begin{bmatrix} J_v \\ J_w \end{bmatrix} = \begin{bmatrix} R_{i-1}^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times (d_n^0 - d_{i-1}^0) \\ R_{i-1}^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix}$$

When calculating 6 degrees of freedom can be found by using the following

$$J_v = [J_{v1}, J_{v2}, J_{v3}, J_{v4}, J_{v5}, J_{v6}]$$

where

$$\begin{aligned} J_{v1} &= K_1 \times (p_{eff} - p_1^B) \\ J_{v2} &= K_2 \times (p_{eff} - p_2^B) \\ &\vdots \\ J_{v6} &= K_6 \times (p_{eff} - p_6^B) \end{aligned}$$

The Jacobian matrix is transformed into a pseudoinverse of the matrix. Pseudoinverse of a matrix is able to work with non-square matrices to invert them, whereas the normal inversion is only capable of interesting squire matrices. Finally, the pseudoinverse matrix is multiplied with the matrix that holds the previous small changes to find the new change in the joint angles.

2.16.5 Communication Protocol

The information being conveyed in digital communication is expressed in digital form, most typically as binary numbers, or bits. Analog data, on the other hand, has a continuous range of values. Most information-transfer systems these days are either digital or in the process of being transformed from analog to digital.

2.13.5.1 Serial Communication

Serial communication is the method of transmitting data. It's similar to the Universal Serial Bus (USB) or Ethernet found in many current Computer systems. Serial communication is used in production facilities to connect various equipment. Serial communication takes the data to be transmitted and sends them one after the other [49]. The process is slower than parallel communication but serial communication generally requires only two connections between devices whereas parallel communication requires more than two connections.

2.13.5.1.1 Universal Asynchronous Receiver-Transmitter (UART)

UART is a full-duplex capable communication protocol that is popular in embedded and desktop systems [28]. This protocol supports peer to peer serial data transfer. Only two connections are required, of which one connection is controlled by one device and another connection is controlled by the other device. Since UART lacks a clock signal to control the data transfer rate, it is predetermined by both devices before transferring data.

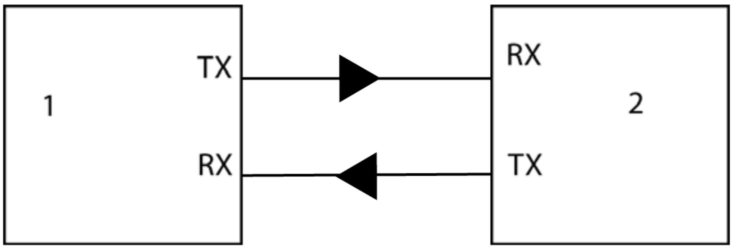

Figure 2.11: UART serial data transfer between 2 devices

2.13.5.1.2 RS485

There are various serial communication protocols, and RS485 is simply one of them. It's well-known for its ability to work across lengthy ranges and in electrically loud surroundings. Because of this, as well as its ability to transfer data over great distances, the RS485 protocol is widely utilized in POS, industry, and telecom applications. In scientific and technological applications, the RS485 is also used in computing, PLCs, embedded systems, and smart sensors [49]. Data is carried by two twisted wires, often known as "Twisted Pair Cable," under the RS485 standard. The twisted pairs in RS485 provide electromagnetic noise immunity, making it suitable for use in electrically loud areas. The two wire setup only allows the RS485 protocol to operate in half-duplex but adding 2 more wires can enable it to operate full-duplex. However, using 4 wire configuration limits the number of devices to be connected to two. In the half-duplex mode, 32 receivers and 32 transmitters can be connected.

2.17 Related Works

Robotics applications have various ways to solve one problem. The six degrees of freedom manipulator is extensively applied to industrial settings. According to [50], in the topic of automatic control, robot supervision is very essential. In the face of a 5% simulation error and disruptions, the operation of a series robot manipulator with adaptable motors was studied. As a study case, a chain manipulator subjected to impact was used to determine the control performance. Furthermore, a pepper harvesting robot named "Sweeper" was developed using 6 DOF manipulators that are able to move using wheels and store the pepper within a container. The robot is able to scan for ripe peppers using deep neural networks and an array of sensors [42]. The picking robot's overall function is discussed, as well as the methods for fruit detection and recognition, grip posture prediction, and motion control. The integrated approach design and validation, as well as thorough field experiments in an industrial greenhouse for various types and growing circumstances, are the key contributions of this work. In [27] the authors discussed covering the gap between cloud and robotic research. Various issues must be addressed, notably ROS and internet improvements, networking, and mobile robots. Because ROS was designed to work in local networks. Another research [15], found very good results on their harvesting robot. Using a perception sensor, the harvesting robot did its work independently. The fruit detection technique was built utilizing a svm with radial basis to recognize and find the apple in the branches autonomously. The gripper and end-effector were guided by the control loop, which included an industrial computer and an AC servo driver, as they approached and plucked the apples. Laboratory studies and outdoor trials in an open area proved the efficiency of the experimental robot system.

Chapter 3

Methodologies and Design

In this section, we are going to discuss how we applied a grid array to find the shortest path to move the robotic system to the desired location, analyze its environment, recognize objects to report or to harvest fruits or vegetables using the six degrees of freedom manipulator mounted on top of the mobile robot. The software part of the robotic system includes various algorithms, multiple deep learning models, robot frameworks, communication protocols, visualization softwares, computer vision and mathematical calculation. On the other hand, the hardware part is again divided into two parts, the x86 architecture based computer and network communication and since mobile robot needs to be able to compute the sophisticated algorithms in order to operate, it has ARM architecture based single board computers, microcontrollers, motor drivers, actuators, actuator position feedback, cameras, wheel motors and batteries. The program of the robotic system runs in a loop from getting the desired grid location to finishing the task. Although detection of objects and performing manipulator tasks are automated by the algorithms, assigning tasks to the robot are done manually at this stage. The entire system uses the ROS framework in one way or another.

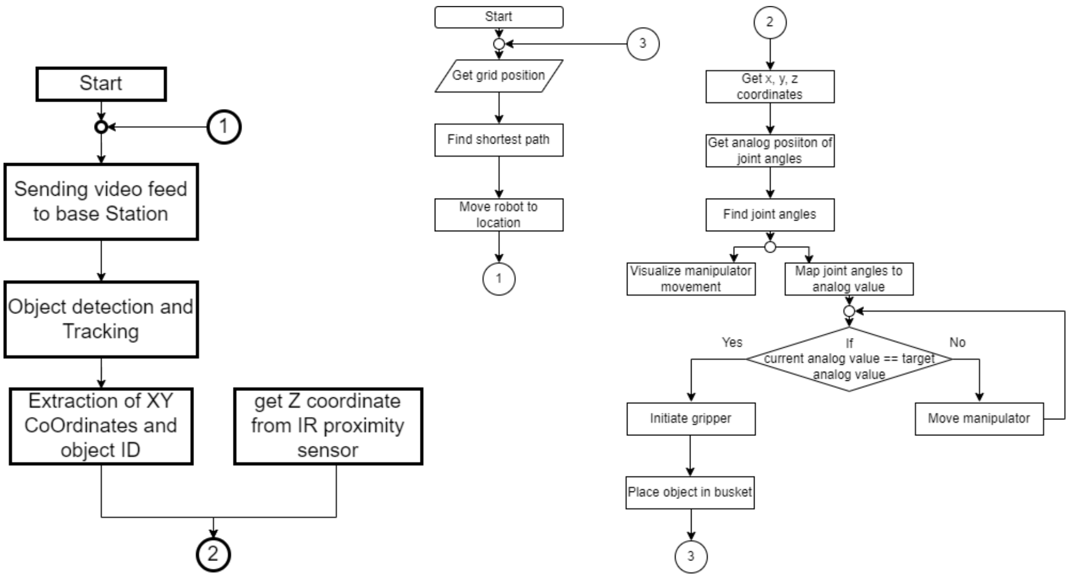

Figure 3.1: Flow chart of the greenhouse robotic system

3.1 Get grid position

Since the robotic system is designed to operate in a greenhouse environment and the plants are to be grown in beds, the robot has to follow predefined lanes to reach locations around the greenhouse. Thus, we have defined the greenhouse layout in a grid array. The user is able to issue the desired location in a graphical user interface.

3.1.1 Graphical User Interface (GUI)

ROS is utilized to create a package that includes nodes that execute the python programs for the GUI which is shown in figure 11. PyQt5 is utilized to design the interface that is used to give grid position commands and if necessary manually control the robot.

3.1.2 ROS network for GUI node

When the ros node is executed, a new node called "robot_controller" spawns. Furthermore, the node is both a publisher and a subscriber. The node uses custom topics and publishes "robot/wheel/motion" and "robot/manipulator/control" topics for manual control of the mobile robot. Moreover, the desired grid position is published to the "robot/grid/moveto" topic from this node. The subscriber part subscribes to the "robot/adc/position" topic. This "robot/adc/position" topic's message contains the positions of the feedback actuators that are displayed in the GUI. Another subscription that the node makes is to the "robot/pwm/set", "robot_controller" node receives the analog values that are to be sent to the manipulator from the SBC. The current position of the mobile robot is sent through the "robot/grid/position" topic, the robot sends messages of the current location of the robot in the grid.

Figure 3.2: GUI of the greenhouse robotic system

3.2 Find Shortest Path

After the user gives the target location, the system has to find the shortest path to the location. There are two popular algorithms that can solve this issue, the Dijkstra's algorithm and the A* algorithm. Both algorithms perform similarly in small scale maps but A* algorithm performs better when using large scale maps [49]. Thus, we decided to use the A* algorithm. The mobile robot receives the target location in the grid via an extension of the main GUI. The extension GUI is developed using the PyGame library. The interface in figure 12 shows the grid array of 800x800, where one meter is equal to 10 grid boxes.

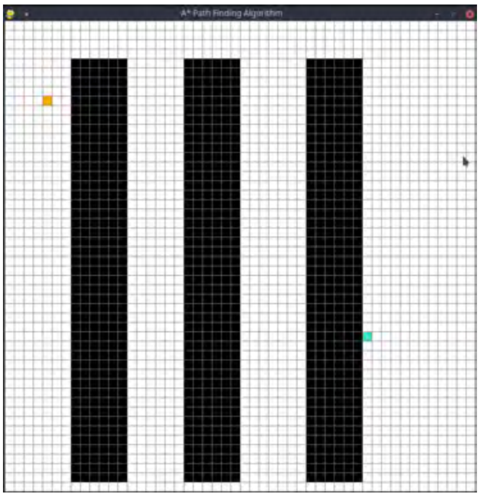

Figure 3.3: 800x800 grid representation of the greenhouse

3.3 Move robot to location

After the shortest path is calculated, the grid data is used to move the robot. The robot moves from one grid to the next getting closer to the target position. The speed of the robot can be manually set but the automatic speed is controlled by the PID algorithm. Specifically, the robot uses proportional and derivative controllers or PD. Integral controller is not used as the robot will not travel long paths and the path will always have straight distances. To tune the PD controller, we used manual tuning.

3.4 Get analog positions of joint angles

Before calculating the joint angle values of the target x, y, z position, we need to know the current position of the x, y, z position of the end-effector. To achieve such, the current joint angles need to be known. In our robot, the joint angles are changed using electric actuators. Each actuator has analog position output that represents the current position of the actuator. We have mapped the analog positions to radians to know the current angles of the joints. The inverse kinematics calculation is done in Radians. Since the chosen single board computer (SBC) does not have analog to digital converters (ADC), we have utilized the Arduino Pro Mini 3.3V 8MHz variant. The reason we choose the 3.3V variant instead of the 5V variant is because most SBCs support 3.3V logic level. The 10 bit ADC of Arduino Pro Mini takes the analog output from the actuators. The 6 degrees of freedom manipulator gives 6 analog values that are inserted to a data packet. The data packet is then sent to the SBC using the UART protocol. UART is one of the most common and easier ways to send ASCII data from one device to another. We have used 115200 baud rate which gives us 52 microsecond latency per bit and 115200 bits/s transfer speed, which is more than enough for our purpose.

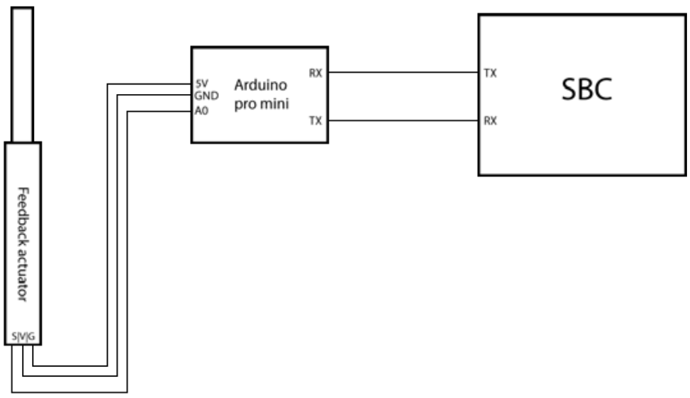

Figure 3.4: Fetching analog values and transmitting it via UART

3.5 Find joint angles

To find the joint angles, we have used inverse kinematics. Forward kinematics is used to find the x, y, z coordinates of the end-effector given the joint angles. But in our case, the deep learning models provide x, y, z position, we need to find the

joint angles. There were various methods to solve the joint angle values. We choose the numerical method, specifically the Pseudo Inverse Jacobian matrix to find the required angles. The analytical approach is dependent on the kinematic diagram of the manipulator. As a result, the approach can not be generalized [43]. Whereas the numerical approach can be generalized and scaled. Although, Pseudo Inverse Jacobian matrix is generalized, to find the current position and goal position, the displacement matrix and rotation matrix of the manipulator is required. This requires the kinematic diagram of the manipulator. Which makes the algorithm partially dependent on kinematic diagrams.
The kinematic diagram of our 6 degree of freedom manipulator is as follows

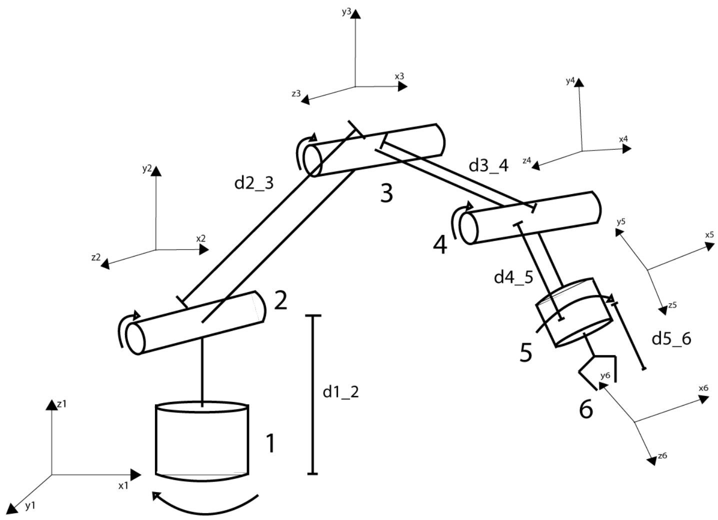

Figure 3.5: Kinematic diagram of the manipulator

We have assumed that the rotation axis of the joints are along the z axis. So, the transition between 1 and 2 in the figure 16 has a 90 degree difference. Furthermore, all the joints are revolute joints. Now we have to find the displacement matrix of the manipulator.

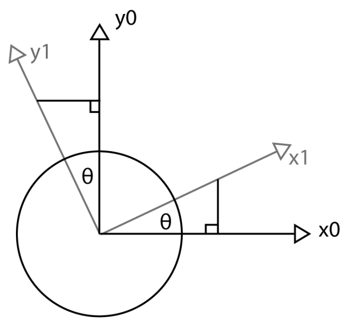

Figure 3.6: Displacement of joint 1

If we view joint 1 along the z axis, we can see that the x and y axis are perpendicular to each other. When the revolute joint rotates along the z axis by degrees, the x and y axis also rotate by degrees and are displaced but the z axis remains in the same direction. So here the new x1, y1 and z1 axis displacement is

$$\begin{bmatrix} d1\_2cos(\theta) \\ d1\_2sin(\theta) \\ d1\_2 \end{bmatrix}$$
$$\begin{bmatrix} d2\_3cos(\theta) \\ d2\_3sin(\theta) \\ d2\_3 \end{bmatrix}$$
$$\begin{bmatrix} d3\_4cos(\theta) \\ d3\_4sin(\theta) \\ d3\_4 \end{bmatrix}$$
$$\begin{bmatrix} d4\_5cos(\theta) \\ d4\_5sin(\theta) \\ d4\_5 \end{bmatrix}$$
$$\begin{bmatrix} d5\_6cos(\theta) \\ d5\_6sin(\theta) \\ d5\_6 \end{bmatrix}$$

As we can see, keeping the rotation axis similar results in the displacement matrix to be similar.
The other necessary component is the rotation matrix. With the displacement matrix, we know where the serial chain needs to go along the axis but we still do not know how much rotation occurred. The rotation matrix consists of two matrices, one is the amount of rotation denoted by sines and cosines and the other is the axis rotation as the z axis is kept as the rotation axis. Matrix (3.3) and 3.4 are multiplied to obtain the rotation matrix of $d1\_2$.

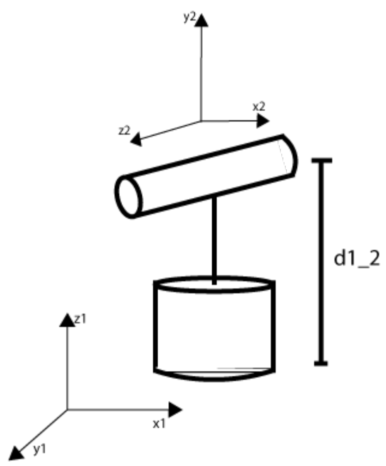

Figure 3.7: d1$_2 rotation$

$$\begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.1}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \tag{3.2}$$

Equation (3.1) is from equation (2.4), as $d1\_2$'s rotation axis is z.
In equation(3.2), the 1 and 2 join's x1 axis are the same so, 1 is placed at the first row and first column of the 3x3 matrix. Similarly, z2 and y1 are in the same direction and z1 and y2 have a matched axis.

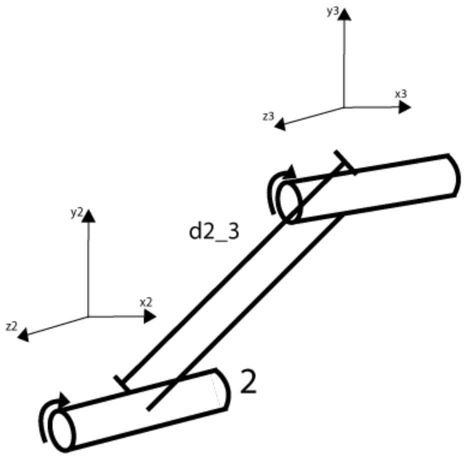

Figure 3.8: d2$_3 rotation$

$$\begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.3}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.4}$$

Again, keeping the z axis as the rotation axis we can use the equation (2.4).
In equation(3.4), x2 and x3, y2 and y3 and z2 and z3 are in the same direction as a result the corresponding 3x3 matrix is as such. Rotation matrix for $d3\_4$ is the same as the rotation matrix of d_3 as the axis and axis displacement are the same.

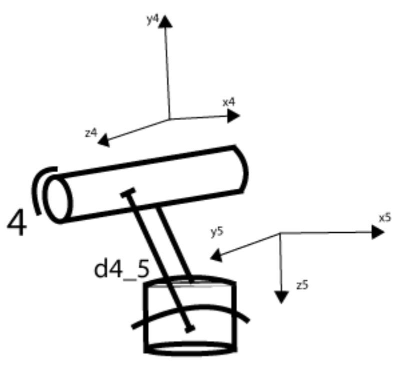

Figure 3.9: d4$_5 rotation$

$$\begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.5}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \tag{3.6}$$

As we can see in figure 18.3, x4 and x5, z4 and y5 are in the same direction and y4 and z5 are in the opposite direction which is represented in matrix (3.6). Rotation matrix of $d4\_5$ is found by multiplying matrix (3.4) and (3.5).
All the rotation matrices are then multiplied to obtain the rotation matrix of the end-effector.
To put together the displacement matrix and rotation matrix, a homogeneous transformation is applied. The homogeneous rotation matrix converts the combined matrices into a 4x4 matrix. Afterwards, the homogeneous rotation matrix is passed into the Pseudo Inverse Jacobian algorithm to find the desired joint angle values in radians.

3.6 Visualize manipulator movement

To obtain a visual representation of the manipulator movement, we have built a URDF model to launch in Gazebo and Rviz. The model consists of a 6 DOF manipulator without the gripper which is placed on top of a 4 wheeled platform. This model represents the physical mobile robot. The goal of this model is to view the kinematics in action.

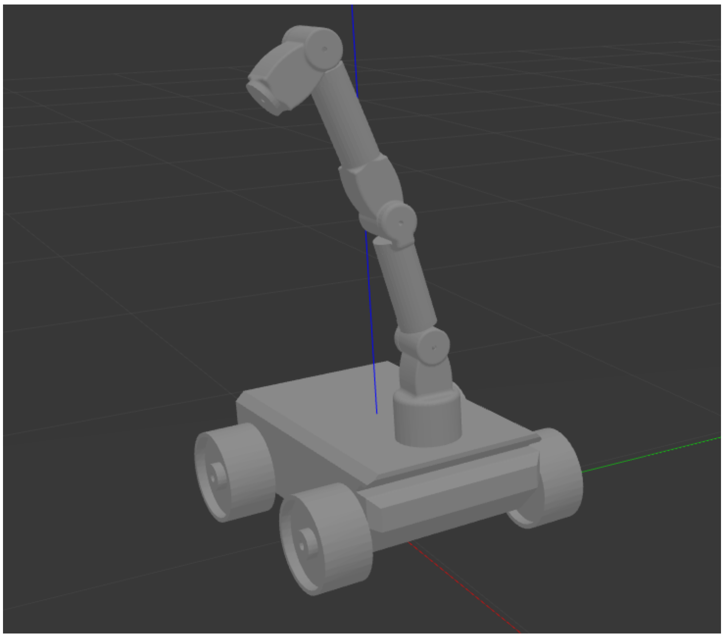

Figure 3.10: Gazebo simulation of the mobile robot

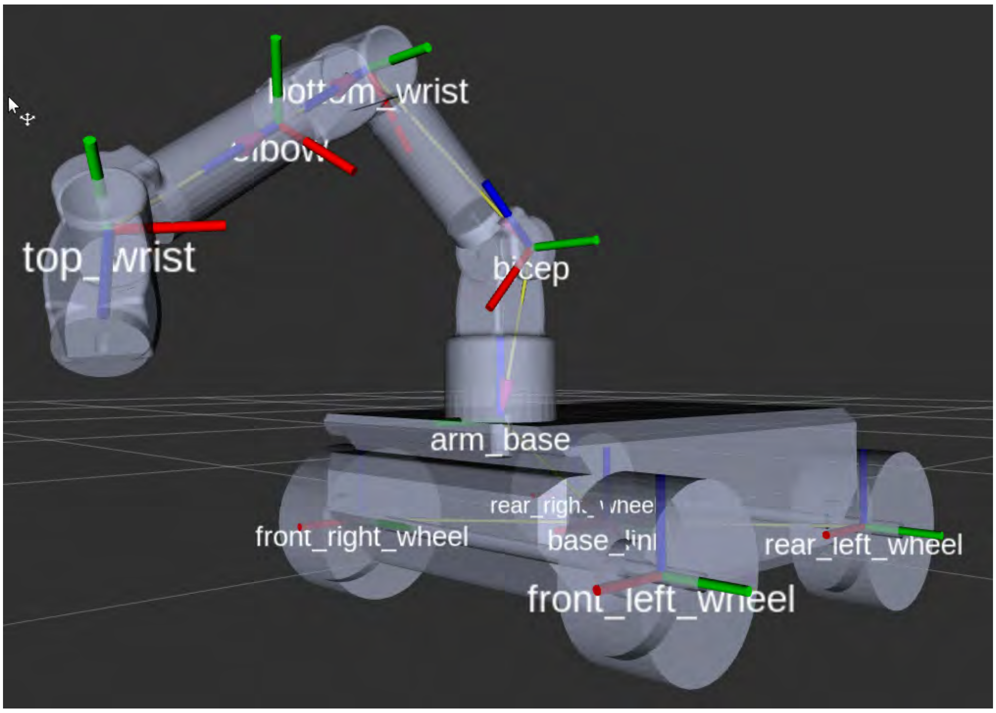

Figure 3.11: RViz simulation of the mobile robot

After the angle values are calculated, the kinematic node publishes them. Visualization programs like Gazebo and RViz subscribes to the topic and move the joints according to the angles. This way, we can evaluate and anticipate the movement of the physical manipulator.

3.7 Map joint angles to analog values

Our physical manipulator will not be able to move by subscribing to the kinematic node's angle values. Since the physical manipulator's joint angles are determined by the analog values of the potentiometers inside the electric actuators. The joint angles are mapped in respect to the ratio of the analog values and published to the robot driver node.

3.8 Move manipulator

The robot driver node compares the current analog values and moves the serial chain of solid bodies until the target analog value is reached. Each time the condition is not met, the joint angles of the manipulator are moved a predetermined amount. The actuators are controlled by VNH2SP30 motor drivers via the single board computer, which provide smooth control to the manipulator.

3.9 Initiate end effector

After reaching the target coordinate, the robotic system checks if it has reached the correct coordinate by comparing with the original target values. A Sharp GP2Y0A51SK0F distance sensor located at the approximate center of the end effector publishes the distance of the object from the end effector and the end effector starts to close. Since fruits or vegetables have different shapes and sizes the tips of the end effector have pressure sensitive switches to stop it from closing any further. After the switches output high signal the system moves on to the next step.

3.10 Place object in basket

The system instructs the manipulator to move to the position where the collection basket is placed. This is the zero position of the manipulator, this eliminates the need to move it to another zero position. The end effector opens up after reaching the target position by changing the joint angles. This does not require finding the angles as they are predetermined. Afterwards the system loops back to get grid position, which initiates either more data collection or harvesting.

3.11 Graph of ROS framework

Since the robotic system uses ROS, each node is connected to each other using topics. The publisher nodes send messages via the topics and subscriber nodes receive the messages by listening to the topic. ROS graph uses a package named "rqt graph" to visualize the relations between the ROS nodes.

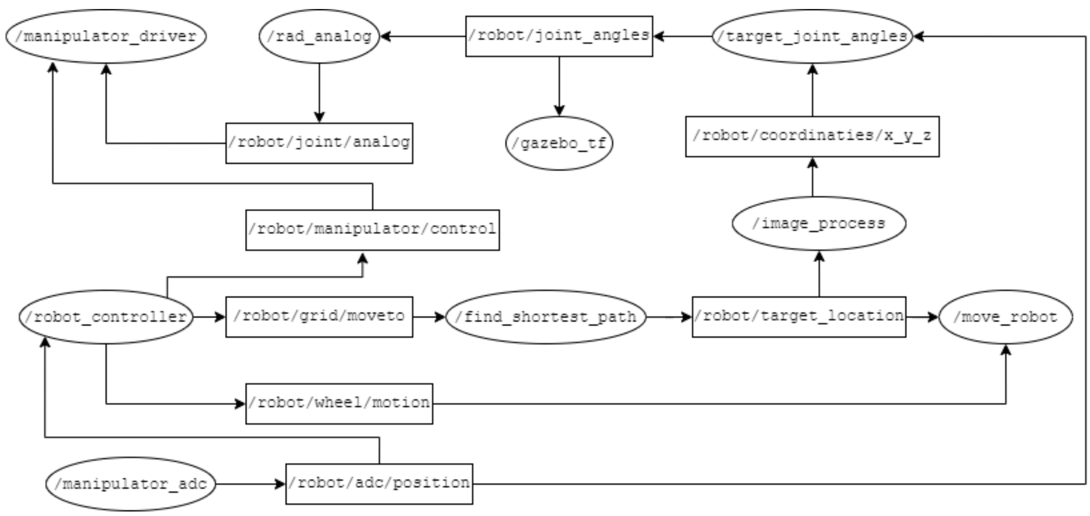

Figure 3.12: rqt graph of the robotic system

3.12 Overview of electronic system

We have designed a circuit on veroboard to connect the components. The overview of the circuit is as follows

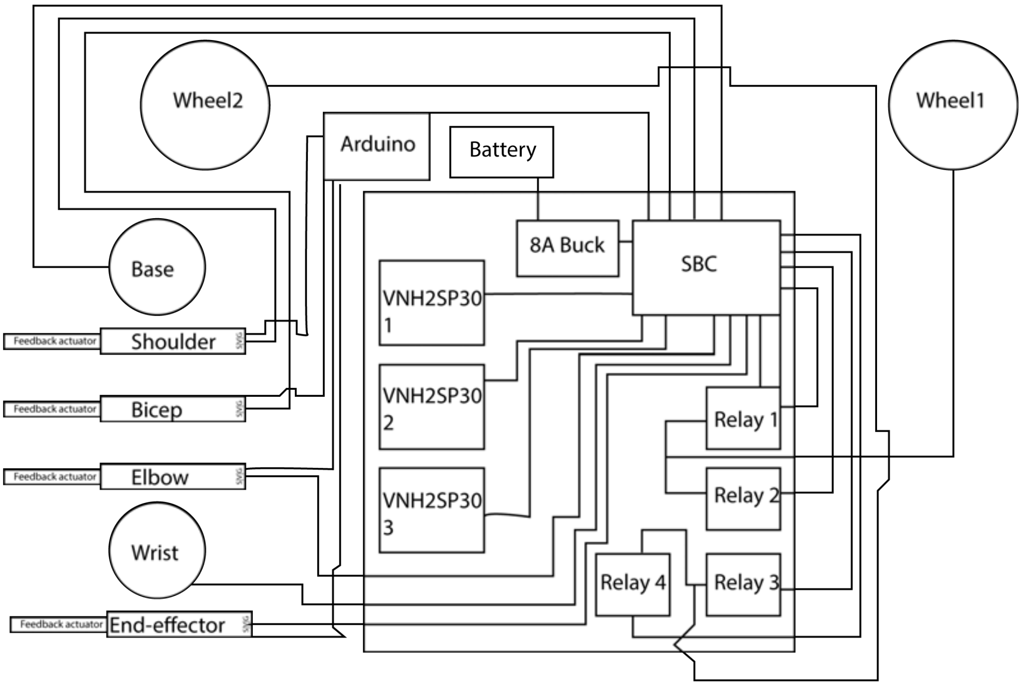

Figure 3.13: Block diagram of the electronic system

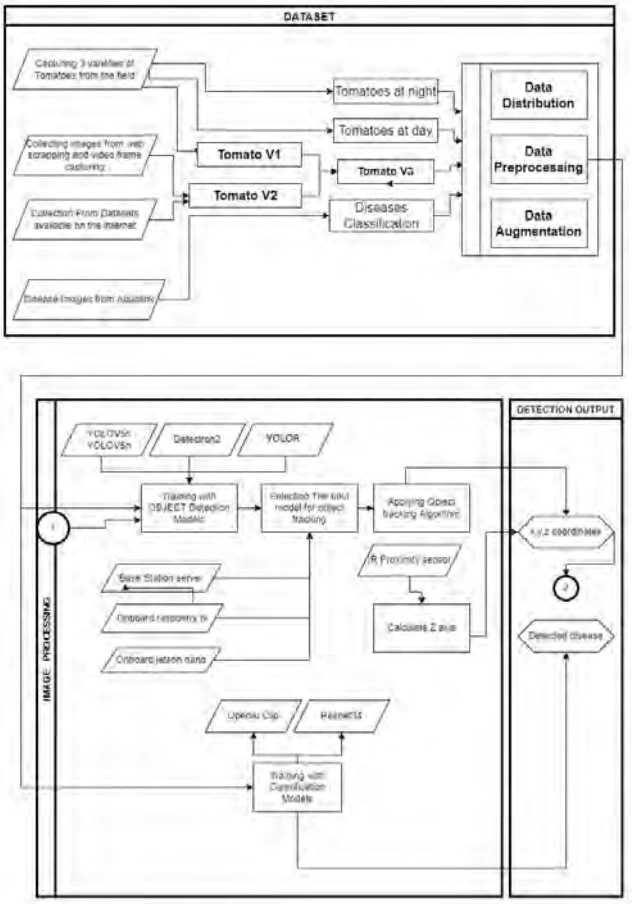

Figure 3.14: Image processing workflow

3.13 Dataset Creation

For training The Object detection and tracking models and classification models we have created our own datasets. We have Three primary datasets in total. There are three classes in all of these datasets as following: ripe, semi ripe and These datasets are briefly explained below:

3.13.1 Tomatoes at night:

This dataset consists primarily of 332 images with 988 instances which we have captured at night .A 50 lumen flashlight helped to lengthen up the objects. The images have been stretched to 416x416 pixels for faster training and running. To overcome the problem of underfitting we have augmented the images in following ways:

1. Bonding boxed Tomatoes has been cropped in a range of 0%-50%
2. Exposure has been applied in a range of -25% to +25% over the Bounding boxed Tomatoes and Images separately.
3. Saturation has been applied in range of -25% to +25% over the Images

4. Images are 90 degree rotated in clockwise counterclockwise and flipped
5. Images has been sheared 15 degrees

After Applying these five augmentation techniques , now we have generated a dataset of 4782 images with a total of 14346 instances.

3.13.2 Tomatoes at day:

850 images have been captured primarily for this dataset in which there are 2803 instances. The images are taken at different times of the day for diversity in ambient light. These images have been stretched to 416x416 pixels for faster training and detection. Image augmentation and preprocessing is the same as The previous dataset.

3.13.3 Tomato V1:

This dataset is consist of 550 images collected using webscrapping and frame capturing from different videos where tomatoes are visible.

3.13.4 Tomato V2:

This is a combined dataset 2 datasets. The first one is "tomatOD" [57].tomatOD is a dataset for tomato fruit localization and ripening classification, containing images of tomato fruits in a greenhouse and high-quality expert annotations from agriculturists. The second one is tomato detection is from kaggle [47].

1. Images:277
   No. of classes:3
   No. of instances:1952
2. Images: 895
   No of classes: 1 (Tomatoes)
   No. of instances: 4,930

3.13.5 Tomato V3:

This version is combination of Tomato V1 and V2. Therefore a total of 1822 images are primarilily here in this dataset and there are 8832 instances.

1. Pre Processing:
   stretched to 416
2. Data Augmentation:
   (a) Cropped 0% - 50%
   (b) Exposure -25% to +25%
   (c) 90 degree rotated in clockwise and anti clockwise
   (d) Sheared 15 degrees

3.13.6 Diseases Classification:

This dataset is collected from Aqualink [53].There are a total of 54, 305 images divided into 38 classes in this dataset. However, for testing our system we have initially used 8 classes of tomato diseases which contain 9978 images. The classes are :

1. Tomato___Bacteria_spot
2. Tomato___Early_blight
3. Tomato___healthy
4. Tomato___Late_blight
5. Tomato___Leaf Mold
6. Tomato___Septoria _leaf_spot
7. Tomato___Target_Spot
8. Tomato___Tomato_mosaic_virus

This dataset is then trained on OpenAi Clip and Resnet34 classification model.

3.14 Model training

For better understanding we have divided this part in two subcategories. One is Object detection model training for tracking the tomatoes and reach to them. For object tracking we have trained on yolov5 small , yolov5 nano , detectron2 and YOLOR

Another one is Tomato diseases detection using classification models.
We have used transfer learning technology for training our custom datasets on deep learning models. We have trained with openAi clip and resnet34 on our Disease classification dataset

3.15 Physical demonstration of the robot

Figure 3.15: Picture of robot

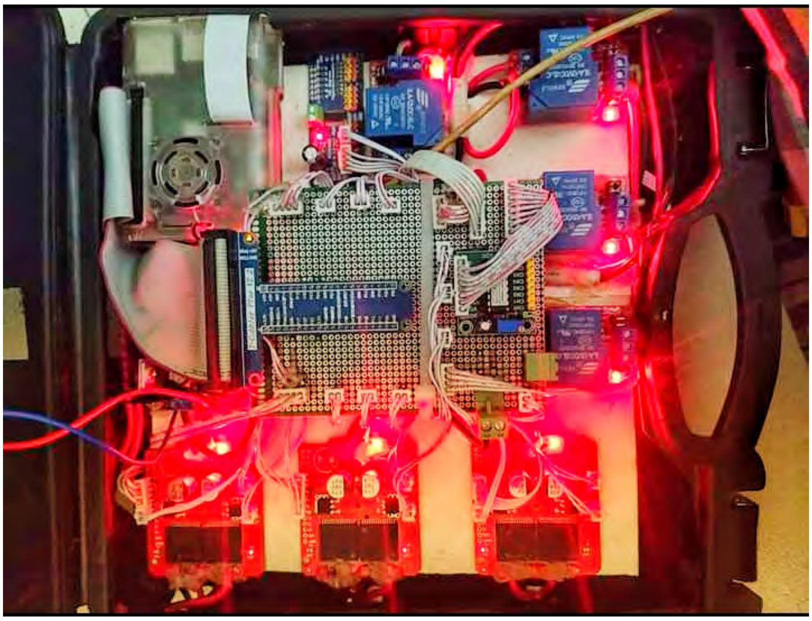

Figure 3.16: Electronics Circuit

In the figures 3.15, 3.16, 3.17, the robot and its circuit is shown. The robot has 2 DC motors with track system. Track system is most used for off road conditions. The manipulator has 6 degrees of freedom, the joints are rotated on their Z axis via feedback electrical linear actuators. The claw is 3 finger design. The robot is powered by lead acid 12v batteries. Furthermore, circuit of the robot has a Raspberry Pi 4 as the main computer, 4 12v relays with maximum current rating of 30A. These relays are used to control the 2 DC motors. The actuators of the

Figure 3.17: Front view of robot

manipulator is controlled by 3 Monster motor drivers. Each of the motor drivers can drive 2 actuators and their peak current voltage is 30A but continuous current of 13A. Moreover, the motor drivers can take 16v. A veroboard is used to construct the circuit board that destributes the Raspberry Pi's 40pins and Arduino pro mini's ADC and PWM pins. Arduino Pro mini 3.3v is used to get the analog values of the feedback actuators. Also, PWM control for the actuators are send via the Arduino.

Chapter 4

Result and Analysis

In this section, With lots of experimentation we found our robot performs at its best at night with the help of its own flashlight. Jetson nano would be used for its better inference speed, portable size and versatility . YoloR with deepsort would be the best fitted model. We analyze the results from the process of implementation. We find how ROS interacts when the system is large and the network has many nodes and topics. Furthermore, the shortest path finding algorithm's performance and usability is discussed. Finding the 3D coordinates of the end effector was quite challenging, the results of the inverse kinematics are shown.

4.1 Comparison between detection at day and night

The amount of images, augmentation and preprocessing techniques and Models being the same , night time dataset performs better in terms of mean average precision and other parameters. Therefore the robot will perform better at night provided a flashlight. This gives us another advantage. It is more efficient and convenient for the robot to work at night when humans can sleep.

Dataset	Model	Algorithm	mAP(%)	GTX 1650	Jetson nano	RPi
At night	Deep Sort	YOLO V5s	67.13	17FPS	3FPS	0.3FPS
At day	Deep Sort	YOLO V5s	69.65	17FPS	3FPS	0.3FPS

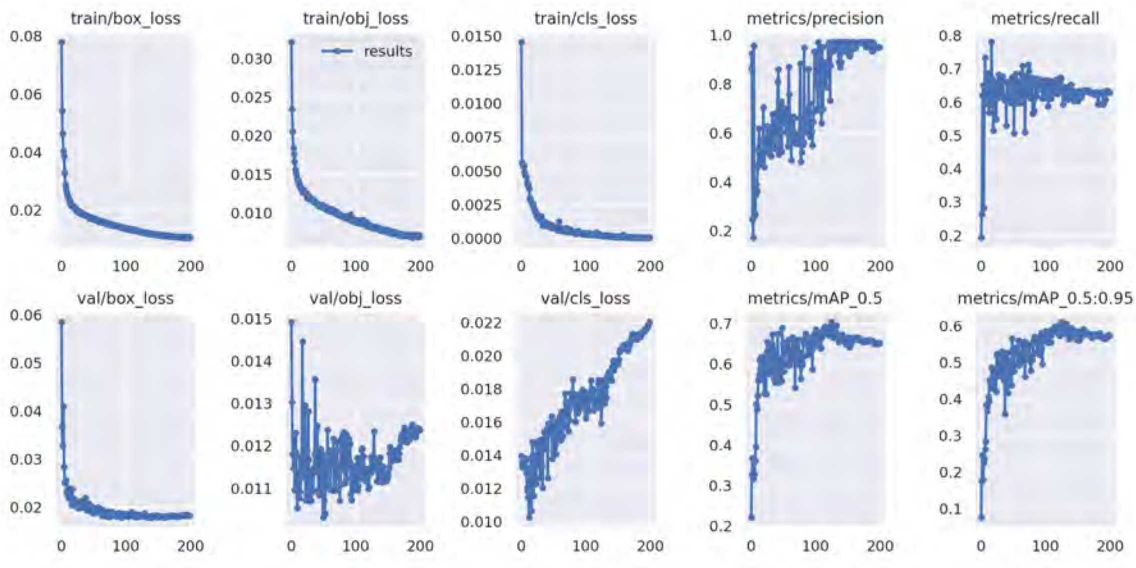

Figure 4.1: Results at day

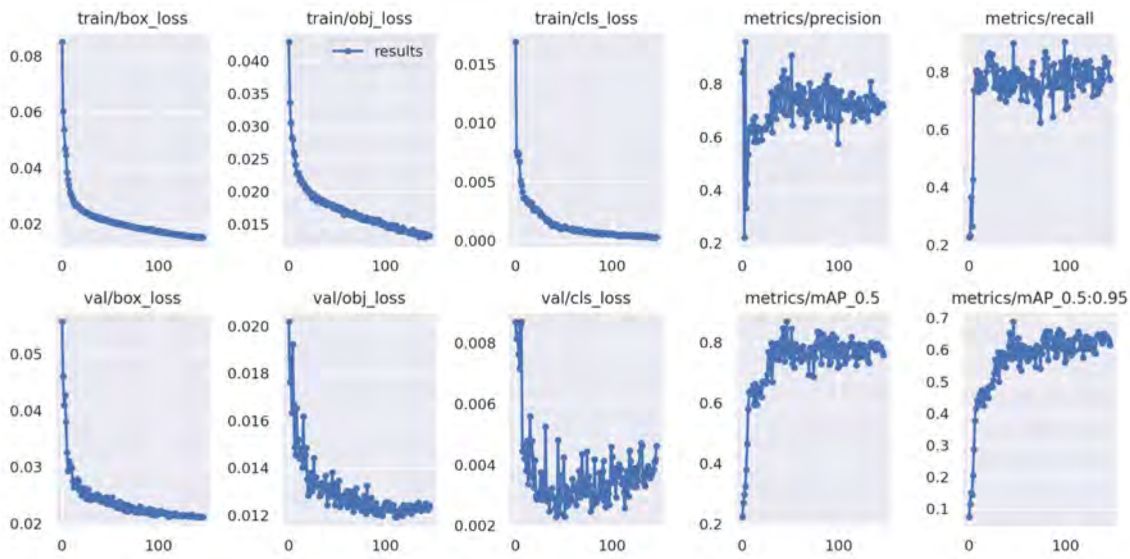

Figure 4.2: Results at night

4.2 Classification Comparison:

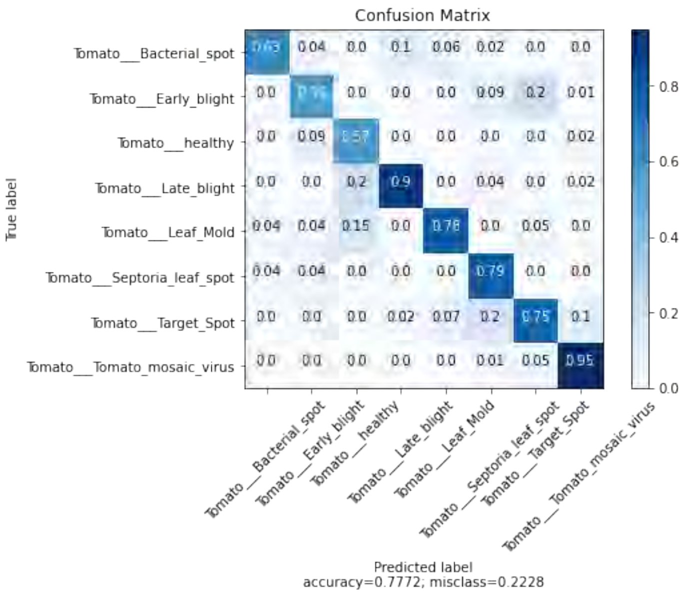

Figure 4.3: Confusion matrix of OpenAI clip

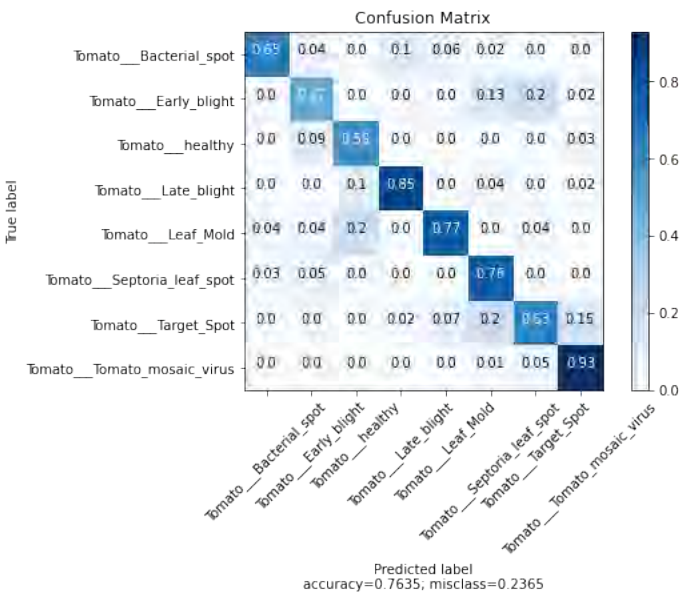

Figure 4.4: Confusion matrix of ResNEt34

4.3 Comparison between object detection models:

Model	mAP	FPS
YOLO V5 n	0.65	4
YOLO V5 s	0.74	3
Detectron2	0.78	2.7
YOLO R	0.85	2.5

Table 4.1: Result analysis of models on jetson nano

4.4 Frame Rate Comparison:

Model	Jetson Nano	Rpi4	GTX1650
YOLOV5n	4FPS	0.7FPS	23FPS
YOLOV5s	3FPS	0.3FPS	17FPS

Table 4.2

4.5 The shortest path to the target position

The latency of sending continuous data over various topics from the controlling laptop to the SBC was high. But After limiting the data sending to once every time data is produced solves the issue.

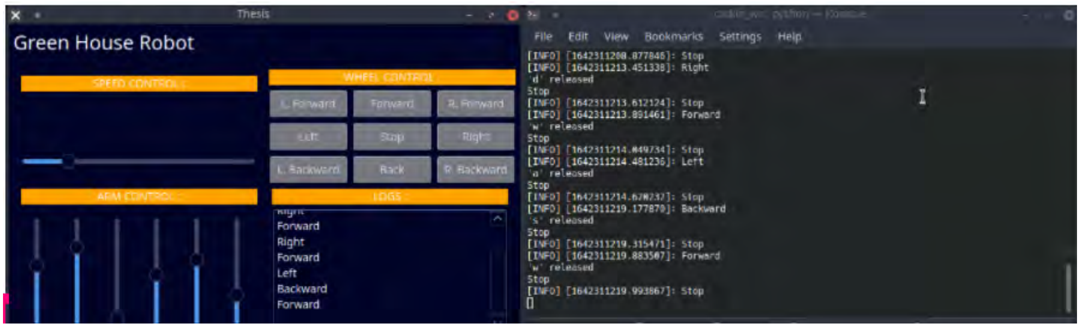

Figure 4.5: ROS Control and feedback of the robotic system

The shortest path to the target position around the grid array using A* algorithm gives the following results.

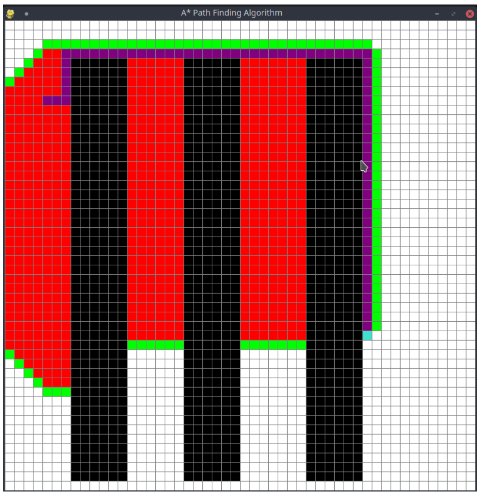

Figure 4.6: Shortest path using A* algorithm of plant bed configuration 1

Figure 4.7: Shortest path using A* algorithm of plant bed configuration 2

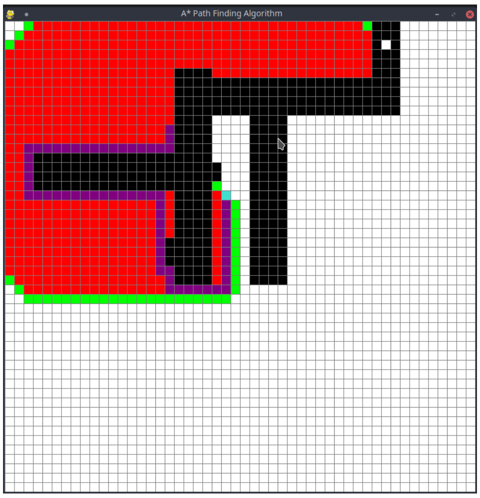

Figure 4.8: Shortest path using A* algorithm of plant bed configuration 3

In figure 4.6, the red boxes indicate the failed attempts that the algorithm had taken. Plant beds are represented by the black boxes that the robot can not go though. The green boxes represent the ongoing path finding areas that were followed. The purple boxes are the shortest path from the current location of the robot to the target position. We have tested multiple grid arrays with various plant bed configurations. The test results we obtained are satisfactory.

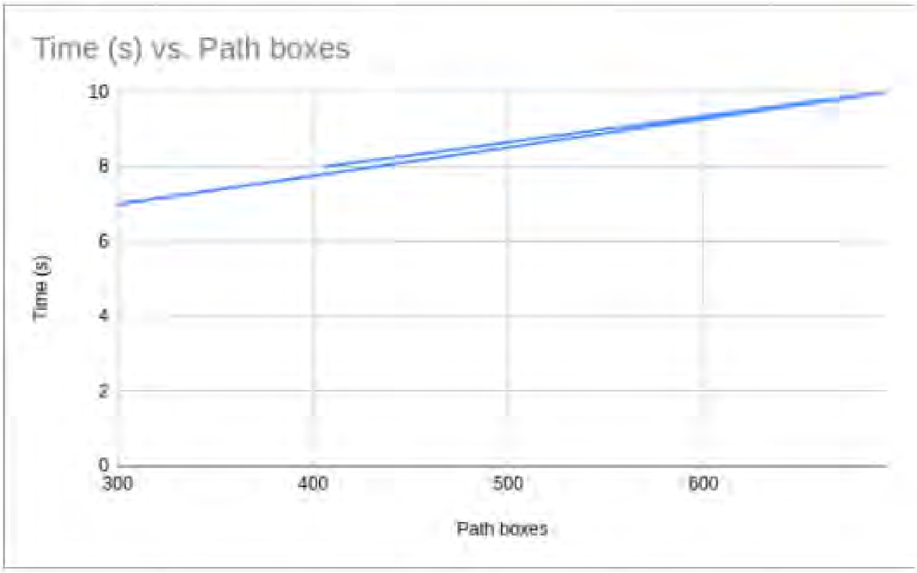

Figure 4.9: Time vs pathbox analysis of A* algorithm

In figure 4.9, we observe that as the path boxes or amount of boxes traversed increase, time to solve the shortest path increases. Compared to figure 23, figure 25 took 2.31% more time.

Since, the electrical actuator provide analog position values, we have to map the analog values to the degrees for each joint. Furthermore, the potentiometers, even when new does not provide the same value for each turn. Thus, a linear relation between the analog values and joint angle values are not possible. We have mapped the degree values according to the changes in the analog values of the potentiomenters. Figure 4.10 shows each relation in a graph from.
The joint angles produced by the Pseudo Inverse of Jacobian are fed to the gazebo URDF model.
Figure 4.11 shows the visualization of the joint angles in Gazebo produced by the kinematic algorithm. As shown before, the Pseudo Inverse of Jacobian takes an iterative approach and calculates the angles in many steps till finding the target position. Figure 4.12 shows the values being calculated by the algorithm to find joint angles in degrees.

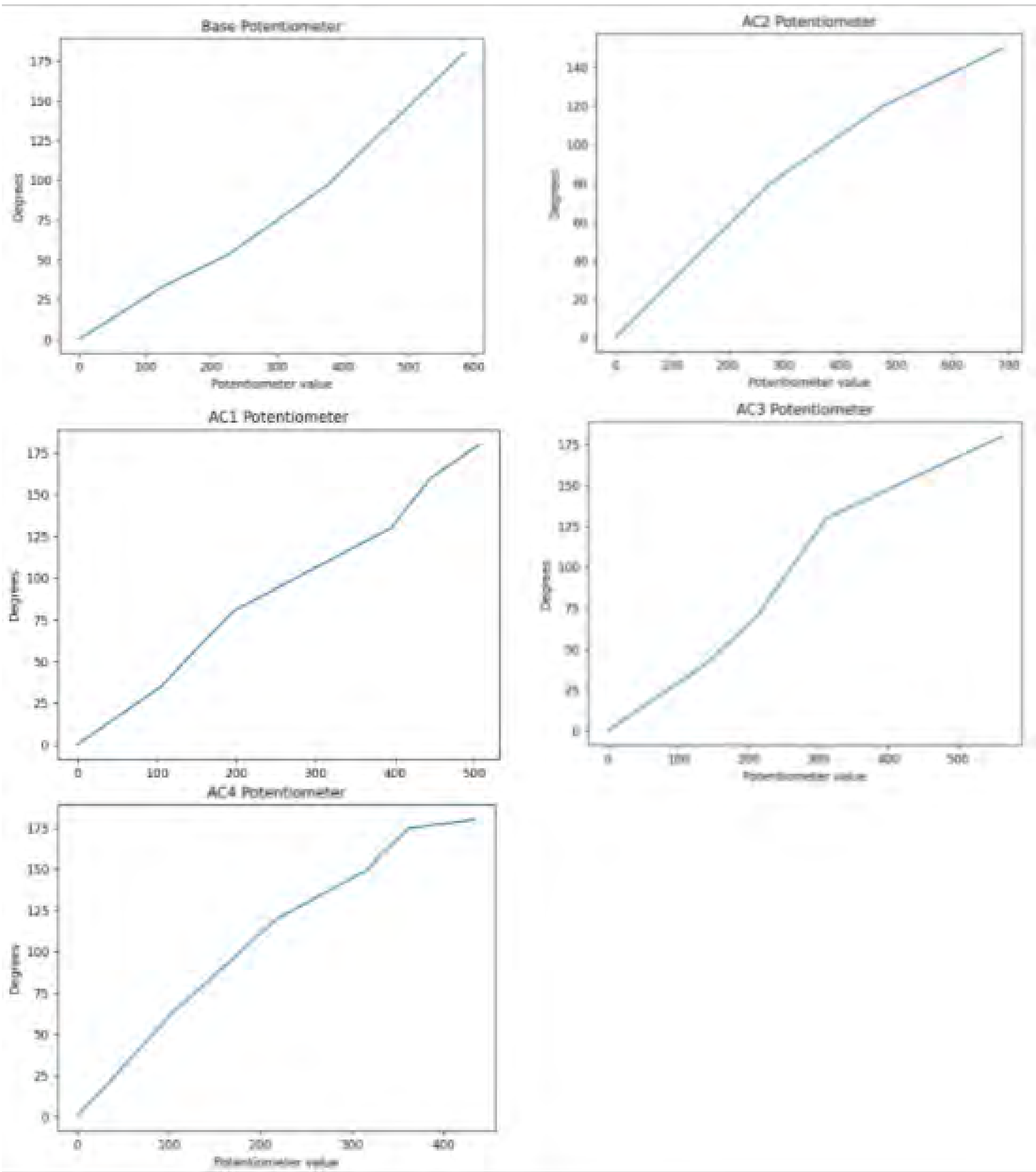

Figure 4.10: Joint degree vs potentiometer value graphs for all joints

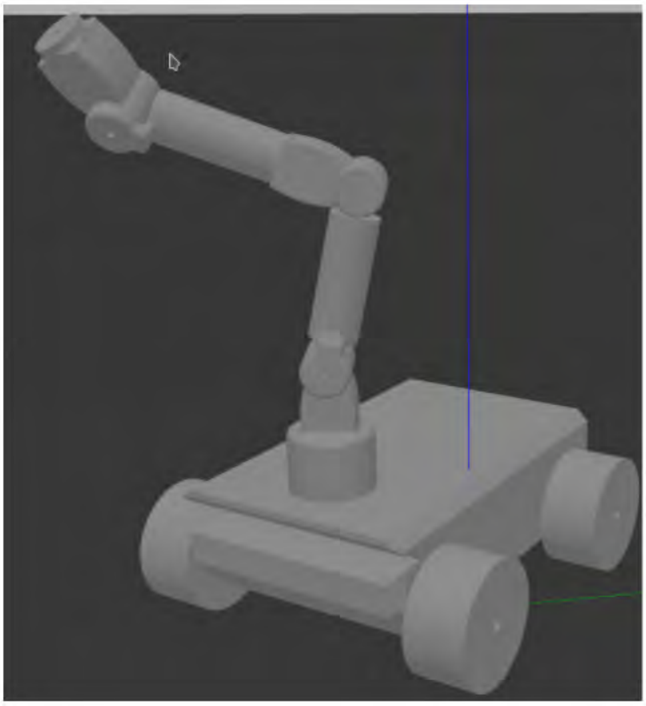

Figure 4.11: Resultant movement of the manipulator

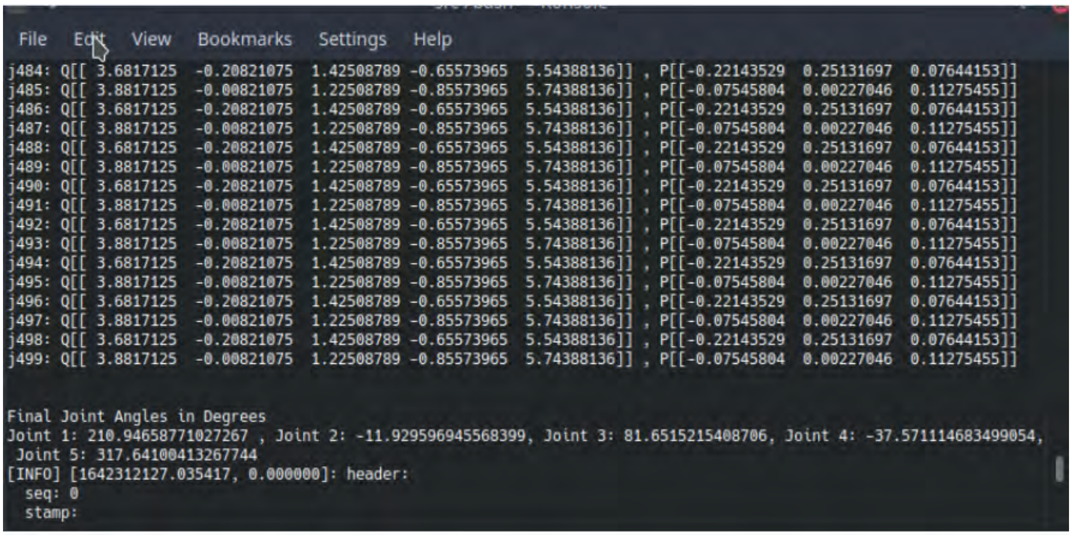

Figure 4.12: Result of Pseudo Inverse of Jacobian

4.6 Discussion

With a lot of trial and error, we discovered that our robot functions best at night with the assistance of its own illumination. Because of its faster inference speed, portable size, and adaptability, the Jetson nano would be employed. The best-fitting model would be YoloR with deepsort .Based on the results of Pseudo Inverse of Jacobian, we can definitely conclude that for the manipulator, this algorithm is best suited. We can not use forward kinematics as our application is not suited for it and the analytical approach to inverse kinematics is not a generalized solution, thus not scalable. Furthermore, the Robot Operating System (ROS) provides the tools and structured communication between nodes. Any system is best designed when the individual programs perform one single task and pass the result to another program.

Chapter 5

Conclusion

5.1 Research Overview

This research work is focused on various algorithms and frameworks to find ways to apply them to the mobile robot system in order to efficiently operate in greenhouse environments and monitor or complete tasks. The whole robotic system is developed on Robot Operating System (ROS) which is a framework that makes robot development more efficient.
Furthermore, in order to move the robot from its current position to the desired position in the greenhouse environment, the greenhouse area is divided into grids. Shortest path finding algorithm is applied to change the position of the mobile robot. In order to monitor or harvest, a manipulator is installed on the top of the mobile robot. We have created our own dataset and experimented with different data augmentation techniques and found the suitable one. We also created datasets of day and night so that we know what time it would be better for the robot to harvest. Then a bunch of pretrained models have been trained using transfer learning for object detection and tracking. We tested the inference time on different processing platforms and presented the Fps.
The manipulator interacts with objects by applying kinematic algorithms to find the target position of the end effector. The robotic system is operated by a python based graphical user interface (GUI) that provides manual and automatic control of the system.

5.2 Research Challenges

Here we are describing the challenges we have faced during the research. We are giving the brief here:

- Implementation of robot operating system on a large scale such as this requires complex interconnection of the framework.
- Algorithms involving shortest paths require a certain amount of time to compute.
- The mobile robot is equipped with ARM based locally available SBCs, thus creating a computation bottleneck.

- Calculating the right kinematics to solve the joint angles for a given end effector coordinate is sensitive as wrong calculations can move the manipulator to the wrong position.
- Electronic logic level compatibility poses an issue that requires careful component selection.
- Collecting and labeling the data was the initial challenge.
- Then we found that different augmentation techniques provoke different training results.
- It was challenging to find the best augmentation techniques for our dataset.
- It takes more resource and time to train various models.

5.3 Experimentation and Results

After training on the tomato v3 dataset, YOLOR gives the most precision while if we consider the inference speed yolov5 nano is the fastest.
Among the classification models OpenAi clip gives better precision than the resNet34. Deep short algorithm slows down the process by around 10 percent.
The shortest path to the target position around the grid array using A* algorithm gives the following results, the red boxes indicate the failed attempts that the algorithm had taken. Plant beds are represented by the black boxes that the robot can not go though. The green boxes represent the ongoing path finding areas that were followed. The purple boxes are the shortest path from the current location of the robot to the target position. We have tested multiple grid arrays with various plant bed configurations. Shortest path using A* algorithm of plant bed configuration.Everytime the algorithm finds the shortest path to the target location but searching for the target takes time before solving the shortest path.We observe that as the path boxes or amount of boxes traversed increase, time to solve the shortest path increases.
The joint angles produced by the Pseudo Inverse of Jacobian are fed to the gazebo URDF model.Shows the visualization of the joint angles in Gazebo produced by the kinematic algorithm. The Pseudo Inverse of Jacobian takes an iterative approach and calculates the angles in many steps till finding the target position.the values being calculated by the algorithm to find joint angles in degrees.

5.4 Contribution and Impact

- We have collected bulk amount of primary data secondary data for training process.
- Compared between some Deep learning models YOLOv5 nano,YOLOv5 small,YOLOR and Detctron2. By this we can find the best fit.
- Compared those Machine Learning models in different devices PC, Laptop and Single Board computer . By this we can estimate the efficient machine.

- CSimulated the robot in Gazebo for better understanding calculations of manipulator position.
- Successfully implemented ROS for operate the robot.
- Successfully detected and tracked Tomato to harvest.
- Completed detection and presented comparison in Day and Night state for real time and non stop tracking.
- Implemented Transfer learning for better detection precision.
- Classified Leaf detection for disease detection for early disease detection.

5.5 Recommendation and Future Work

The main advice for this suggested system is to deal with the difficulties it confronts when there are other cars on the road. Another will be raising the amount of precision during any unpredictability and performing preferred action in accordance with reality. The object classifier, on the other hand, has a lot of room for improvement in terms of decreasing the average amount of misclassifications and raising the overall mean of accuracy levels. By constructing a confusion matrix, you can easily see how much accuracy has been gained through these steps. To tackle the misclassification problem, we can boost the iteration score when training object classes.

Bibliography

[1] A. Goldenberg, B. Benhabib, and R. Fenton, "A complete generalized solution to the inverse kinematics of robots," *IEEE Journal on Robotics and Automation*, vol. 1, no. 1, pp. 14–20, 1985.

[2] G. H. Brundtland, M. Khalid, *et al.*, *Our common future*. Oxford University Press, Oxford, GB, 1987.

[3] F. Merat, "Introduction to robotics: Mechanics and control," *IEEE Journal on Robotics and Automation*, vol. 3, no. 2, pp. 166–166, 1987.

[4] K. Nelson, W. Cunningham, R. Andersen, G. Harrison, and L. Gelberg, "Is food insufficiency associated with health status and health care utilization among adults with diabetes?" *Journal of General Internal Medicine*, vol. 16, no. 6, pp. 404–411, 2001.

[5] S. Kucuk and Z. Bingul, *Robot kinematics: Forward and inverse kinematics*. INTECH Open Access Publisher, 2006.

[6] J.-P. Merlet, "Jacobian, manipulability, condition number, and accuracy of parallel robots," 2006.

[7] G. Fischer, F. N. Tubiello, H. Van Velthuizen, and D. A. Wiberg, "Climate change impacts on irrigation water requirements: Effects of mitigation, 1990–2080," *Technological Forecasting and Social Change*, vol. 74, no. 7, pp. 1083–1107, 2007.

[8] J. Kramer and M. Scheutz, "Development environments for autonomous mobile robots: A survey," *Autonomous Robots*, vol. 22, no. 2, pp. 101–132, 2007.

[9] H. K. Seligman, A. B. Bindman, E. Vittinghoff, A. M. Kanaya, and M. B. Kushel, "Food insecurity is associated with diabetes mellitus: Results from the national health examination and nutrition examination survey (nhanes) 1999–2002," *Journal of general internal medicine*, vol. 22, no. 7, pp. 1018–1023, 2007.

[10] P. K. Streatfield and Z. A. Karar, "Population challenges for bangladesh in the coming decades," *Journal of health, population, and nutrition*, vol. 26, no. 3, p. 261, 2008.

[11] A. Aristidou and J. Lasenby, "Inverse kinematics: A review of existing techniques and introduction of a new fast iterative solver," 2009.

[12] J. J. Craig, *Introduction to robotics: mechanics and control, 3/E*. Pearson Education India, 2009.

[13] M. Quigley, K. Conley, B. Gerkey, *et al.*, "Ros: An open-source robot operating system," in *ICRA workshop on open source software*, Kobe, Japan, vol. 3, 2009, p. 5.

[14] H. K. Seligman, B. A. Laraia, and M. B. Kushel, "Food insecurity is associated with chronic disease among low-income nhanes participants," *The Journal of nutrition*, vol. 140, no. 2, pp. 304–310, 2010.

[15] Z. De-An, L. Jidong, J. Wei, Z. Ying, and C. Yu, "Design and control of an apple harvesting robot," *Biosystems engineering*, vol. 110, no. 2, pp. 112–122, 2011.

[16] W. Bank, *World development report 2012: Gender equality and development*. The World Bank, 2011.

[17] J. A. Foley, N. Ramankutty, K. A. Brauman, *et al.*, "Solutions for a cultivated planet," *Nature*, vol. 478, no. 7369, pp. 337–342, 2011.

[18] M. Phillips, M. Beveridge, F. Weirowski, W. Rogers, A. Padiyar, *et al.*, "Financing smallholder aquaculture enterprises.," *WorldFish Center Policy Brief*, no. 2011-07, 2011.

[19] J. Kerr and K. Nickels, "Robot operating systems: Bridging the gap between human and robot," in *Proceedings of the 2012 44th Southeastern Symposium on System Theory (SSST)*, IEEE, 2012, pp. 99–104.

[20] T. Garnett, M. C. Appleby, A. Balmford, *et al.*, "Sustainable intensification in agriculture: Premises and policies," *Science*, vol. 341, no. 6141, pp. 33–34, 2013.

[21] D. Halliday, R. Resnick, and J. Walker, *Fundamentals of physics*. John Wiley & Sons, 2013.

[22] O. A. Ibironke, "Glasshouse production of vegetables and ornamentals for agricultural productivity in nigeria," *World J. Agric. Sci*, vol. 1, pp. 113–119, 2013.

[23] B. Ghose, B. Razib, and G. Sharmistha, "Reviewing the status of agricultural production in bangladesh from a food security perspective," *Russian journal of agricultural and socio-economic sciences*, vol. 25, no. 1, 2014.

[24] C. Hedley, J. Knox, S. Raine, and R. Smith, "Water: Advanced irrigation technologies," 2014.

[25] Y. Lu and D. Liu, "Kinematics analysis and simulation on transfer robot with six degrees of freedom," *Sensors & Transducers*, vol. 176, no. 8, p. 285, 2014.

[26] P. Alexander, M. D. Rounsevell, C. Dislich, J. R. Dodson, K. Engstrøm, and D. Moran, "Drivers for global agricultural land use change: The nexus of diet, population, yield and bioenergy," *Global Environmental Change*, vol. 35, pp. 138–147, 2015.

[27] K. Alisher, K. Alexander, and B. Alexandr, "Control of the mobile robots with ros in robotics courses," *Procedia Engineering*, vol. 100, pp. 1475–1484, 2015.

[28] M. Awal, "Vulnerability to disaster: Pressure and release model for climate change hazards in bangladesh," *International journal of environmental monitoring and protection*, vol. 2, no. 2, pp. 15–21, 2015.

[29] H. R. Kam, S.-H. Lee, T. Park, and C.-H. Kim, "Rviz: A toolkit for real domain data visualization," *Telecommunication Systems*, vol. 60, no. 2, pp. 337–345, 2015.

[30] S. Motala, S. Ngandu, S. Mti, *et al.*, "Millennium development goals: Country report 2015," 2015.

[31] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS: a practical introduction to the Robot Operating System.* " O'Reilly Media, Inc.", 2015.

[32] W. Wu and B. Ma, "Integrated nutrient management (inm) for sustaining crop productivity and reducing environmental impact: A review," *Science of the Total Environment*, vol. 512, pp. 415–427, 2015.

[33] A. Yousuf, C. C. Lehman, M. A. Mustafa, and M. M. Hayder, "Introducing kinematics with robot operating system (ros)," in *2015 ASEE Annual Conference & Exposition*, 2015, pp. 26–1024.

[34] B. Siciliano and O. Khatib, *Springer handbook of robotics.* springer, 2016.

[35] K. Benke and B. Tomkins, "Future food-production systems: Vertical farming and controlled-environment agriculture," *Sustainability: Science, Practice and Policy*, vol. 13, no. 1, pp. 13–26, 2017.

[36] S. Fiaz, M. A. Noor, and F. O. Aldosri, "Achieving food security in the kingdom of saudi arabia through innovation: Potential role of agricultural extension," *Journal of the Saudi Society of Agricultural Sciences*, vol. 17, no. 4, pp. 365–375, 2018.

[37] M. K. Hasan, S. Desiere, M. D'Haese, and L. Kumar, "Impact of climate-smart agriculture adoption on the food security of coastal farmers in bangladesh," *Food Security*, vol. 10, no. 4, pp. 1073–1088, 2018.

[38] L. Joseph and J. Cacace, *Mastering ROS for Robotics Programming: Design, build, and simulate complex robots using the Robot Operating System.* Packt Publishing Ltd, 2018.

[39] J. F. Velasco-Muñoz, J. A. Aznar-Sánchez, L. J. Belmonte-Ureña, and M. J. López-Serrano, "Advances in water use efficiency in agriculture: A bibliometric analysis," *Water*, vol. 10, no. 4, p. 377, 2018.

[40] J. F. Velasco-Muñoz, J. A. Aznar-Sánchez, L. J. Belmonte-Ureña, and I. M. Román-Sánchez, "Sustainable water use in agriculture: A review of worldwide research," *Sustainability*, vol. 10, no. 4, p. 1084, 2018.

[41] J. A. Aznar-Sánchez, M. Piquer-Rodrıguez, J. F. Velasco-Muñoz, and F. Manzano-Agugliaro, "Worldwide research trends on sustainable land use in agriculture," *Land use policy*, vol. 87, p. 104 069, 2019.

[42] B. Arad, J. Balendonck, R. Barth, *et al.*, "Development of a sweet pepper harvesting robot," *Journal of Field Robotics*, vol. 37, no. 6, pp. 1027–1039, 2020.

[43] A. automaticaddison, *How to find displacement vectors for robotic arms*, Aug. 2020. [Online]. Available: https://automaticaddison.com/how-to-find-displacement-vectors-for-robotic-arms/.

[44] A. Caradja, "Evaluation of the quality of the outputs of the vocational education system with agricultural profile based on its contribution to the achievement of the tasks for sustainable rural areas development," 2020.

[45] S. Fernando, R. Nethmi, A. Silva, A. Perera, R. De Silva, and P. K. Abeygunawardhana, "Intelligent disease detection system for greenhouse with a robotic monitoring system," in *2020 2nd International Conference on Advancements in Computing (ICAC)*, IEEE, vol. 1, 2020, pp. 204–209.

[46] O. Kasatkina, C. Masclet, J.-F. Boujut, and E. de Vries, "Format effects in the understanding of motion from kinematic diagrams in engineering education," *International Journal of Technology and Design Education*, pp. 1–17, 2020.

[47] Larxel, *Tomato detection*, Jun. 2020. [Online]. Available: https://www.kaggle.com/andrewmvd/tomato-detection.

[48] A. A. Süzen, B. Duman, and B. Şen, "Benchmark analysis of jetson tx2, jetson nano and raspberry pi using deep-cnn," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, IEEE, 2020, pp. 1–5.

[49] D. O. 1. m. ago, D. Ong, D. O. a. posts, and S. a. posts, *All about rs485 - how rs485 works and how to implement rs485 into industrial control systems?* Jul. 2021. [Online]. Available: https://www.seeedstudio.com/blog/2021/03/18/how-rs485-works-and-how-to-implement-rs485-into-industrial-control-systems/.

[50] K. S. Gaeid, A. F. Nashee, I. A. Ahmed, and M. H. Dekheel, "Robot control and kinematic analysis with 6dof manipulator using direct kinematic method," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 1, pp. 70–78, 2021.

[51] A. Takimoglu, *What is data augmentation? techniques, benefit examples*, Nov. 2021. [Online]. Available: https://research.aimultiple.com/data-augmentation/.

[52] *What is data preprocessing? - definition from techopedia*, Jul. 2021. [Online]. Available: https://www.techopedia.com/definition/14650/data-preprocessing.

[53] [Online]. Available: https://aqualinkbangladesh.com/.

[54] D. Lippman, *Mathematics for the liberal arts*. [Online]. Available: https://courses.lumenlearning.com/waymakermath4libarts/chapter/shortest-path/.

[55] Osrf, *Why gazebo?* [Online]. Available: http://gazebosim.org/.

[56] K. Protocol, "Targets for the first commitment period," in *United Nations Framework Convention on Climate Change Available at: http://unfccc. int/kyoto_protocol/items php.(Accessed: 4th May 2016).*

[57] up2metric, *Up2metric/tomatod: Tomatod dataset for tomato fruit localizatin and ripening classification.* [Online]. Available: https://github.com/up2metric/tomatOD.

[58] *What is the a* algorithm?* [Online]. Available: https://www.educative.io/edpresso/what-is-the-a-star-algorithm.

[59] *Wiki*. [Online]. Available: http://wiki.ros.org/ROS/TCPROS.