# ELECTROHUB : WIRED AUTONOMOUS DOCKING SYSTEM FOR ELECTRIC VEHICLES
# 18MHP109L - MAJOR PROJECT REPORT

***Submitted by***


**Priyanshu Ranka (RA2011018010033)**
**Keshav Bhardwaj (RA2011018010038)**
**Devyanshu Khare (RA2011018010052)**
**(Batch No.: MHP 40)**


*Under the guidance of*


**Dr. Anitha Kumari, M.E., Ph.D.**
(Assistant Professor, Department of Mechatronics Engineering)


*In Partial fulfilment for the degree*

*of*

**BACHELOR OF TECHNOLOGY**

in

**MECHATRONICS ENGINEERING (Mention specialization if any)**

of

**COLLEGE OF ENGINEERING & TECHNOLOGY**


SRM INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

S.R.M. Nagar, Kattankulathur, Kancheepuram District


**May 2024**

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Under Section 3 of UGC Act, 1956)

# BONAFIDE CERTIFICATE

Certified that this project report titled "*Electrohub : Autonomous Docking System for Electric Vehicles*" is the bonafide work of "**Priyanshu Ranka (RA2011018010033), (RA2011018010038) and Devyanshu Khare (RA2011018010052)**, who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

13/5/24

SIGNATURE

Dept. of Mechatronics Engineering
Kattankulathur
603 203
* SRM IST *

13/5/24

SIGNATURE

**Dr. Anitha Kumari, M.E., Ph.D.**
GUIDE
Assistant Professor
Dept. of Mechatronics Engineering

**Dr. G. MURALI, M.E., Ph.D.**
PROFESSOR & HEAD OF THE DEPARTMENT
Dept. of Mechatronics Engineering

14/5/2024

Signature of the Internal Examiner

14/5/2024

Signature of the External Examiner

# ABSTRACT

Electrohub presents a groundbreaking initiative in the realm of electric vehicle (EV) charging infrastructure—a wired autonomous docking system designed to streamline the charging process. The project entails the integration of robotics, computer vision, and automation technologies to develop an efficient and user-friendly solution. In this prototype, a robotic arm with a charger as its end effector is mounted on a moving vehicle, facilitating seamless charging experiences for EV owners.

Key features of Electrohub include the utilization of machine learning algorithms for vehicle and charging port detection, as well as precise positioning of the charger. A sophisticated camera system, coupled with Raspberry Pi 4 as the primary microcontroller, enables real-time image processing and analysis. The robotic arm, powered by 995MG and Orange OT5320M servo motors, exhibits exceptional control and accuracy in maneuvering. Additionally, the arm's robust metal body ensures stability during docking and undocking procedures.

Electrohub's innovative approach significantly reduces human intervention in the charging process. By leveraging machine learning for vehicle and charging port detection, the system autonomously identifies compatible EVs and accurately locates their charging ports. This capability not only streamlines the charging experience for users but also enhances operational efficiency and reliability. With Electrohub, EV owners can enjoy hassle-free charging experiences, thereby accelerating the adoption of sustainable transportation solutions.

# ACKNOWLEDGEMENTS

It has been a great honor and privilege to undergo **B.Tech** in **MECHATRONICS** at **SRM Institute of Science and Technology**. We are very much thankful to the **Department of Mechatronics, SRM Institute of Science and Technology** for providing all facilities and support to meet our project requirements.

We would like to thank our Head of Department **Dr. G. MURALI, M.E., Ph.D.** and our project guide **Dr. Anitha Kumari, M.E., Ph.D.** who gave us a big helping hand during the entire project.

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely fortunate that we have got this all along the completion of our project work. Whatever we have done is only due to such guidance and assistance and we would like to thank them for their kind support.

**PRIYANSHU RANKA**

**KESHAV BHARDWAJ**

**DEVYANSHU KHARE**

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER-1

## 1.1 BACKGROUND OF THE PROJECT

In tandem with the rise of Electric Vehicles (EVs), the establishment of comprehensive EV charging infrastructure is paramount. Charging stations play a pivotal role in alleviating range anxiety among EV drivers by providing convenient access to charging facilities, thereby fostering widespread adoption of electric transportation. They enable longer journeys, making EVs a practical choice for both urban commuters and long-distance travelers. Furthermore, expanding the charging network promotes equitable access to clean transportation, ensuring that individuals from diverse socioeconomic backgrounds can participate in the electric revolution. Additionally, investing in charging infrastructure stimulates economic growth by creating jobs in construction, maintenance, and related industries. Moreover, strategically locating charging stations in public spaces, workplaces, and residential areas encourages the integration of EVs into daily routines, further accelerating the transition towards sustainable mobility. Thus, EV charging stations are indispensable for realizing the full potential of electric vehicles and advancing towards a greener, more resilient transportation ecosystem.

### 1.1.1 Plug-in Charging for EVs

The most common way for electric vehicles (EVs) to recharge their batteries is through plug-in charging. It works in the same way as when you charge any electronic device. Technically speaking, an Electric Vehicle Supply Equipment (EVSE) cable is what connects the charging outlet on your automobile to a power source. Fig. 1.1 demonstrates traditional plug-in charging, where the plug needs to be connected to the car. Charging time could go from 30 minutes to 2 hours . Depending on the kind of EVSE and the vehicle's specifications, there are many charging speed options. For overnight charges with shorter commutes, Level 1 charging uses a regular household outlet. Faster charging times are provided by level 2 charging stations, which are frequently located in public areas and houses with suitable installation. Lastly, DC fast chargers offer the fastest way, boosting range quickly and significantly, making them perfect for lengthy journeys. But normally, these can be found at special charging stations.

**Figure 1.1: Plug-In Charging in EVs**

### 1.1.2 Problems with Charging EVs

Electric vehicle (EV) adoption faces multifaceted challenges, including infrastructure limitations, charging speed concerns, and hardships related to the physical handling of charging. Firstly, the availability and accessibility of charging infrastructure remain significant hurdles. Rural areas often lack charging stations, and urban centers may not have enough to meet demand. This disparity can cause range anxiety, exacerbated by differing standards and connectors among stations, leading to confusion and inconvenience for EV owners. Secondly, while fast-charging technology exists, most stations offer slower speeds, resulting in long wait times and disrupted travel plans. Additionally, the cost of charging and billing complexities further deter adoption, with inconsistent pricing models and billing errors eroding trust in the system.

Moreover, physical handling challenges add to the EV charging burden. Maneuvering cables in adverse weather conditions proves cumbersome and sometimes hazardous, especially for those with limited mobility. Accessibility issues at public stations compound these difficulties, as drivers may struggle to access charging ports due to design flaws or wear and tear. Addressing these challenges requires concerted efforts to expand infrastructure coverage, improve charging speeds, and streamline the physical charging process. Innovations such as wireless charging and user-friendly station designs can enhance the EV ownership experience, making it more accessible and convenient for all drivers.

### 1.1.3 Wireless charging as an alternative to wired charging

Wireless charging for electric vehicles eliminates the need for cumbersome cables by using electromagnetic fields to transfer energy between a charging pad on the ground and a receiver coil in the vehicle. This approach offers unmatched convenience, allowing drivers to simply park over a designated pad for automatic and contactless charging. Particularly beneficial for those with mobility challenges or in adverse weather conditions, wireless charging streamlines the charging process and promotes seamless integration of electric vehicles into daily routines. Although still in early stages, ongoing advancements aim to enhance efficiency and affordability, promising a user-friendly solution to traditional wired charging methods.

### 1.1.4 Advantages of Wired over wireless charging

While wireless charging for electric vehicles offers convenience and ease of use, traditional wired charging methods have several advantages. Firstly, wired charging typically provides faster charging speeds compared to wireless alternatives, allowing EV owners to replenish their battery more quickly. Additionally, wired charging systems are generally more energy-efficient, resulting in lower energy losses during the charging process. Moreover, wired charging infrastructure is more mature and widely available, offering greater coverage and accessibility for EV owners. Finally, wired charging solutions tend to be more cost-effective than wireless alternatives, making them a more economical choice for both consumers and charging station operators. Despite the convenience of wireless charging, these advantages highlight the continued relevance and importance of wired charging options in the electric vehicle ecosystem.

### 1.1.5 Types of Wired Charging

#### 1. Level 1 Charging

Level 1 charging is the simplest and most basic form of EV charging. It involves using a standard household electrical outlet, typically rated at 120 volts AC, and a portable charging cable that comes with the EV. While convenient for home charging, Level 1 charging is slow, providing only a few miles of range per hour of charging. It's suitable for overnight charging when the vehicle is parked for an extended period, such as at home or at work.

#### 2. Level 2 Charging

Level 2 charging utilizes a higher-powered 240-volt AC charging station, which can be installed at home or in public locations such as parking garages, shopping centers, or workplaces. These stations offer faster charging speeds compared to Level 1, typically providing around 10-60 miles of range per hour, depending on the vehicle and charging station specifications. Level 2 charging is suitable for daily charging needs and is commonly used for home charging and destination charging.

#### 3. DC Fast Charging (DCFC)

DC Fast Charging, also known as Level 3 charging, is the fastest form of charging available for electric vehicles. These high-powered DC charging stations are typically located along highways and in commercial areas, enabling rapid charging for long-distance travel. DC Fast Chargers can deliver a significant amount of energy to the battery in a short time, providing up to 80% of the battery's capacity in around 30 minutes. This makes them ideal for quick pit stops during road trips or when drivers need to recharge on the go.

4. Tesla Supercharging

Tesla vehicles utilize a proprietary charging network called Superchargers, which are high-speed DC charging stations specifically designed for Tesla EVs. Superchargers can deliver up to 250 kW of power, allowing Tesla drivers to recharge their vehicles quickly and efficiently at dedicated Tesla charging stations located strategically along major travel routes. Tesla Superchargers are ideal for long-distance travel and are often used by Tesla owners for cross-country road trips.

Charging in India

In India, Level 2 charging stations are the most common, with a growing number of public charging infrastructure installations in urban areas. DC Fast Charging stations are also being deployed along major highways to support long-distance travel. While Tesla Superchargers are not prevalent in India due to Tesla's limited presence in the market, other manufacturers and charging infrastructure providers are working to expand the charging network across the country to support the growing adoption of electric vehicles.

## 1.2 NEED FOR THE PROJECT

Convenience, efficiency, time-saving, scalability, technological advancement are few of the many ways in which the idea for Electrohub can be spelled out. Eliminating the need for human intervention, making charging EVs more convenient and safer. Electrohub shows integration of robotics and automation in a simple yet effective way for everyday tasks.

### 1.3 OBJECTIVE

Electrohub presents a groundbreaking initiative in the realm of electric vehicle (EV) charging infrastructure—a wired autonomous docking system designed to streamline the charging process. The project entails the integration of robotics, computer vision, and automation technologies to develop an efficient and user-friendly solution. In this prototype, a robotic arm with a charger as its end effector is mounted on a moving vehicle, facilitating seamless charging experiences for EV owners.

Key features of Electrohub include the utilization of machine learning algorithms for vehicle and charging port detection, as well as precise positioning of the charger. A sophisticated camera system, coupled with Raspberry Pi 4 as the primary microcontroller, enables real-time image processing and analysis. The robotic arm, powered by 995MG and Orange OT5320M servo motors, exhibits exceptional control and accuracy in maneuvering. Additionally, the arm's robust metal body ensures stability during docking and undocking procedures.

Electrohub's innovative approach significantly reduces human intervention in the charging process. By leveraging machine learning for vehicle and charging port detection, the system autonomously identifies compatible EVs and accurately locates their charging ports. This capability not only streamlines the charging experience for users but also enhances operational efficiency and reliability. With Electrohub, EV owners can enjoy hassle-free charging experiences, thereby accelerating the adoption of sustainable transportation solutions.

# CHAPTER-2

# LITERATURE SURVEY

### 2.1 A Review Of Robotic Charging For Electric Vehicles

The paper by Hendri Maja Saputra, Nur Safwati Mohd Nor, Intan Zaurah Mat Darus, and Edwar Yazid provides a comprehensive review of the current state of research on robotic charging systems for EVs. The authors examine the technical aspects, research trends, methods, and challenges associated with the development of autonomous charging solutions for EVs.

The review covers various aspects of robotic charging, including the use of 3D vision guidance systems to detect and navigate towards the charging ports of EVs, as demonstrated in the work by Mišeikis et al. The authors also discuss the integration of depth sensors, such as the ZED 2i sensor, to enable accurate detection and localization of the charging sockets, as explored in the study by Zhang and Jin. Additionally, the review explores the advancements in cable-driven manipulators and machine vision-based algorithms for fast and reliable identification and localization of the charging ports, as presented in the research by Lou et al. and Quan et al.

The literature review highlights the potential benefits of robotic charging systems, including improved user convenience, increased charging efficiency, and reduced environmental impact through the promotion of electric vehicle adoption. The authors also discuss the technical challenges and research gaps, such as the need for robust and reliable detection algorithms, collision avoidance strategies, and seamless integration with the charging infrastructure.

Overall, this comprehensive literature review provides valuable insights into the current state of the art in robotic charging for electric vehicles, and serves as a useful reference for researchers and practitioners working in this rapidly evolving field.

### 2.2 Docking Method For Electric Vehicles Charging Terminal Using Monocular Camera

The paper by Donghee Noh, Seonhyeong Kim, Seojeong Kim, Donghoon Kim, Kyoungho Choi, Juhwan Choi, and Keunho Park presents a novel docking method for electric vehicle charging terminals using a monocular camera.

The authors recognize the importance of developing efficient and user-friendly charging solutions to support the widespread adoption of electric vehicles. To address this, they propose a docking method that leverages a monocular camera to detect and guide the vehicle towards the charging terminal, eliminating the need for complex sensor setups or manual intervention.

The key aspects of the proposed docking method include:

1. Monocular camera-based detection and localization of the charging terminal: The authors develop a computer vision-based algorithm to detect and localize the charging terminal using a single camera, without the need for additional depth sensors.

2. Docking guidance and control: The system provides real-time feedback to the vehicle's control system, guiding it towards the optimal docking position for seamless charging.

3. Robustness and reliability: The authors address challenges such as varying lighting conditions and occlusions to ensure the docking method is reliable and can operate in diverse real-world scenarios.

The literature review highlights the advantages of this monocular camera-based docking approach, including its simplicity, cost-effectiveness, and potential for widespread adoption in the electric vehicle charging infrastructure. The authors also discuss the technical considerations and future research directions to further enhance the performance and usability of the proposed system.

Overall, this literature review provides valuable insights into the development of user-friendly and efficient docking solutions for electric vehicle charging, leveraging advanced computer vision techniques to improve the charging experience and support the transition towards sustainable transportation.

### 2.3 3D Vision Guided Robotic Charging Station For Electric And Plug-In Hybrid Vehicles

The paper titled "3D Vision Guided Robotic Charging Station for Electric and Plug-in Hybrid Vehicles" by Justinas Mišeikis, Matthias Rüther, Bernhard Walzel, Mario Hirz, and Helmut Brunner presents a novel approach to the charging of electric vehicles using robotics and 3D vision technology. The system is designed to automate the charging process, enhancing efficiency and convenience for electric vehicle owners.

The paper begins by highlighting the challenges associated with the charging of electric vehicles, including the need for a reliable and accessible charging infrastructure. The authors then introduce their proposed solution, which involves the use of a robotic charging station guided by 3D vision technology. This system is capable of autonomously detecting and navigating towards the charging ports of electric vehicles, ensuring a precise and efficient charging process.

The paper provides a detailed overview of the system's architecture, including the 3D vision system, the UR10 robot, and the charging station. The authors also discuss the shape-based matching methods used to identify and locate the charging port, as well as the three-step robot motion planning procedure for plug-in charging.

The paper concludes by presenting the results of experiments conducted to demonstrate the functionality of the system. The results show that the system is capable of successfully identifying and charging electric vehicles, highlighting the potential for this technology to improve the efficiency and convenience of electric vehicle charging.

In terms of the broader context of the paper, it is part of a growing body of research focused on the development of autonomous charging systems for electric vehicles. This research is driven by the need for efficient and user-friendly charging infrastructure to support the widespread adoption of electric vehicles.

The paper's findings are significant because they demonstrate the potential for 3D vision technology to enhance the efficiency and convenience of electric vehicle charging. The use of robotics and 3D vision technology can help to automate the charging process, reducing the need for human intervention and improving the overall user experience.

In terms of the implications of the paper's findings, they suggest that the development of autonomous charging systems like the one presented in the paper could play a key role in the widespread adoption of electric vehicles. By providing efficient and user-friendly charging infrastructure, these systems can help to alleviate range anxiety and make electric vehicles more appealing to consumers.

Overall, the paper presents a novel approach to the charging of electric vehicles using robotics and 3D vision technology. The system is capable of autonomously detecting and navigating towards the charging ports of electric vehicles, ensuring a precise and efficient charging process. The paper's findings are significant because they demonstrate the potential for 3D vision technology to enhance the efficiency and convenience of electric vehicle charging, and the implications of the paper's findings suggest that the development of autonomous charging systems like the one presented in the paper could play a key role in the widespread adoption of electric vehicles.

### 2.4 Research on Fast Identification and Location of Contour Features of Electric Vehicle Charging Port in Complex Scenes

The research paper "Research on Fast Identification and Location of Contour Features of Electric Vehicle Charging Port in Complex Scenes" by Pengkun Quan et al. presents a comprehensive approach to developing a vision-guided system for accurately identifying and locating the charging ports of electric vehicles in diverse environmental conditions.

The key aspects of the research methodology highlighted in the paper include:

1. Data Acquisition and Preprocessing: The researchers collected an extensive dataset of charging port images under varying lighting conditions (indoor/outdoor, sunny/cloudy, day/night) to ensure the robustness of their algorithms. They also implemented an automatic exposure control algorithm to maintain consistent image brightness.

2. Rough Positioning: For the initial rough positioning stage, the researchers employed Hough circle and Hough line detection algorithms to identify the circular and linear features of the charging port. This allowed them to roughly locate the charging port within the camera's field of view.

3. Precise Positioning: Building upon the rough positioning results, the researchers used Canny edge detection and the QCS (Quadratic Curve Segment) method to accurately fit ellipses to the contours of the charging port. They then utilized the PNP (Perspective-n-Point) algorithm to determine the precise 3D pose of the charging port relative to the camera.

4. Experimental Evaluation: The researchers thoroughly evaluated their system's performance under diverse environmental conditions, reporting high success rates for both rough and precise positioning tasks. An overall average success rate of 94.8%.

The key insights and methodologies presented in this paper can be leveraged for various research and project work in related domains:

1. The robust data acquisition and preprocessing techniques, including the use of automatic exposure control, can be applied to develop vision-guided systems for other applications that require reliable object detection and localization in complex environments.

2. The combination of Hough-based feature detection and ellipse fitting using the QCS method offers a versatile framework for accurately identifying and localizing various types of objects with distinct geometric features.

3. The integration of the PNP algorithm for 3D pose estimation provides a valuable reference for researchers working on vision-based robotic manipulation and navigation tasks.

4. The detailed experimental evaluation and analysis of performance under different lighting and weather conditions can guide the development of adaptive and resilient computer vision algorithms for real-world applications.

Overall, the research presented in this paper contributes significantly to the field of vision-guided systems for electric vehicle charging, while also providing a solid foundation for future research and project work in related areas of computer vision, robotics, and automation.

### 2.5 AViTRoN: Advanced Vision Track Routing and Navigation for Autonomous Charging of Electric Vehicles

The research paper "AViTRoN: Advanced Vision Track Routing and Navigation for Autonomous Charging of Electric Vehicles" presents a comprehensive framework for enabling autonomous charging of EVs through the integration of advanced vision-based technologies.

A key aspect of the research that can be leveraged for other projects is the utilization of a Siamese lightweight hourglass network called SATIN. This network is designed to capture contextual information at multiple scales, tackling the challenge of efficient and robust object detection and localization. The cross-attentional module within SATIN boosts the discriminative and localization capabilities of the system, making it a valuable tool for various vision-guided applications.

Another notable contribution is the emphasis on predictive maintenance facilitated by sophisticated artificial intelligence models. The paper discusses how these models analyze historical usage data and employ predictive analytics to identify subtle patterns and potential issues within the charging infrastructure. The proactive maintenance schedules triggered by these predictive insights can significantly reduce downtime and enhance the overall reliability of charging networks, a concept that can be adapted to other domains requiring predictive maintenance.

The research also explores sustainable solutions, such as solar-powered charging stations equipped with energy storage systems. These stations operate independently from the grid, harnessing solar energy and minimizing environmental impact. The intricate energy management algorithms developed for these stations can be valuable for researchers working on renewable energy integration and microgrid applications.

Furthermore, the paper delves into the evolution of wireless charging technologies, including magnetic resonance and inductive coupling. These advancements in wireless charging can inspire researchers to explore similar solutions for improving convenience and accessibility in various domains beyond electric vehicle charging.

The integration of vehicle-to-grid (V2G) technology, which enables bidirectional energy flow between the grid and EV batteries, is another area that can be explored in other energy-related projects. The concept of EVs supplying excess energy back to the grid during peak demand periods can be adapted to broader energy management and grid optimization strategies.

In terms of the technical implementation, the paper provides insights into the use of the YOLOv8 object detection model, which has demonstrated exceptional performance on benchmark datasets like COCO. The adaptability and versatility of YOLOv8 can be leveraged in various computer vision-based projects that require accurate object recognition and localization.

Additionally, the paper's discussion of the Robot Operating System (ROS) and its modular architecture, including packages like actionlib, nodelet, and rosbridge, offers a valuable reference for researchers and developers working on complex robotic systems. The ROS framework's emphasis on collaboration, reusability, and efficient communication can be applied to a wide range of robotics and automation projects.

Overall, the research presented in this paper not only contributes to the field of autonomous charging for electric vehicles but also provides a wealth of insights and methodologies that can be adapted and applied to various other research and project work in related domains, such as computer vision, predictive maintenance, renewable energy integration, and robotic software development.

## 2.6 Autonomous Charging Docking Control Method for Unmanned Vehicles Based on Vision and Infrared

The research paper "Autonomous Charging Docking Control Method for Unmanned Vehicles Based on Vision and Infrared" presents a comprehensive approach to enabling autonomous charging for unmanned vehicles through the integration of vision-based and infrared-guided technologies.

A key aspect of the research that can be leveraged for other projects is the design of the autonomous charging device and intelligent charging stand. The authors have developed a system that combines a camber-type electric core, charging adsorption device, and autonomous charging docking mechanism. This modular design approach can be adapted to various robotic and autonomous systems that require reliable and efficient charging solutions.

Another notable contribution is the two-stage charging docking method, which utilizes a combination of vision-based rough positioning and infrared-guided precise positioning. This hybrid approach addresses the challenges of accurately aligning the unmanned vehicle with the charging stand, a critical requirement for seamless and reliable charging. The vision-based rough positioning, which relies on lane detection and vehicle centering, can be applied to other autonomous navigation and docking tasks beyond electric vehicle charging.

Furthermore, the paper's emphasis on safety features, such as the anti-touch and obstacle avoidance algorithms, provides a valuable reference for researchers working on the development of secure and reliable autonomous systems. The authors' approach to detecting potential hazards, including the use of pressure sensors and distance measurement modules, can be adapted to enhance the safety of various robotic applications.

The communication and control modules designed for the integration between the unmanned vehicle and the charging stand also offer insights for researchers working on coordinated multi-agent systems. The authors' implementation of Bluetooth-based communication and the integration of various sensors and actuators can serve as a blueprint for developing robust and scalable control architectures.

Additionally, the experimental evaluation and the reported success rates for charging docking and obstacle avoidance provide valuable data points for researchers to benchmark their own work. The authors' attention to performance metrics, such as docking time and pressure differences, can guide the development of optimization strategies for autonomous systems.

Overall, the research presented in this paper not only contributes to the field of autonomous charging for unmanned vehicles but also provides a wealth of insights and methodologies that can be adapted and applied to various other research and project work in related domains, such as robotic navigation, docking systems, safety-critical control, and multi-agent coordination.

# CHAPTER-3

# HARDWARE IMPLEMENTATION

## 3.1 DESIGN

### 3.1.1 Design of the robotic arm

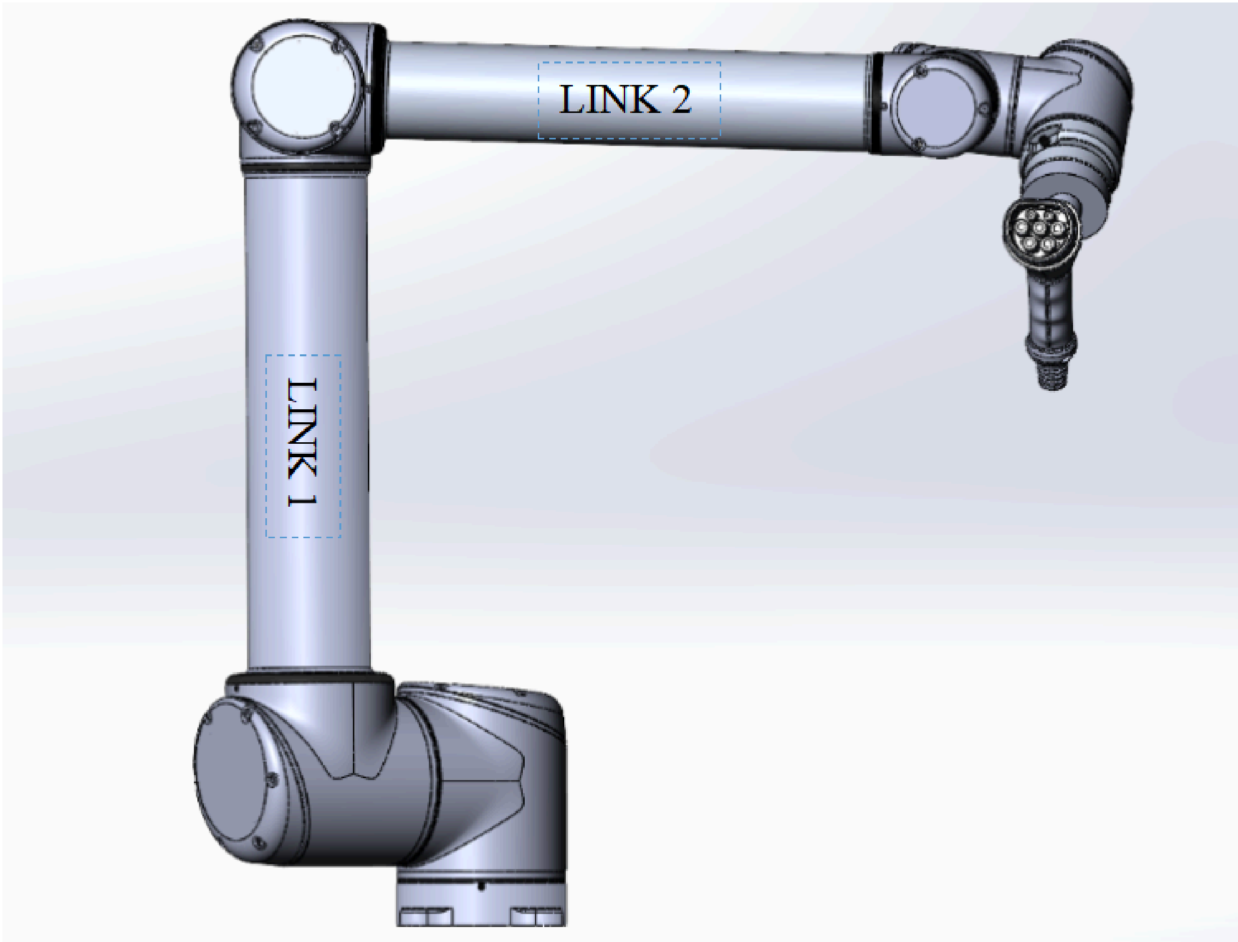
The following figures depict the design of the final product intended for the market, engineered to be suitable for installation and utilization in real-world scenarios and to meet market demands. This design iteration was informed by an extensive review of research papers featuring analogous products, drawing inspiration from various designs and selecting the most fitting design that aligns with our project requirements. The primary requirement was a 6 Degrees of Freedom (DOF) robotic arm capable of providing sufficient maneuverability to reach the vehicle from any orientation, while ensuring stability. The UR10 robotic arm best fulfilled these criteria, offering the necessary versatility and stability required for the task. The UR10 excels in executing repetitive tasks with precision, thereby mitigating safety risks. Its adaptability to changes in production processes translates to reduced labor costs and enhanced productivity. As a highly flexible and programmable robot, the UR10 is capable of performing a diverse array of tasks. Consequently, it emerged as an optimal solution for enhancing productivity and profitability while prioritizing safety. We tailored a similar design to meet our specific requirements, adjusting link lengths to achieve the desired reach and modifying the end-effector to accommodate the EV charger plug. This manipulator will be mounted on an elevation platform for optimal functionality.

The primary specification dictated the need for a 6 Degrees of Freedom (DOF) robotic arm capable of reaching a distance of 1.6 meters while maintaining stability. To fulfill this requirement, we selected the UR10 robotic arm, renowned for its versatility, stability, and precision. With its 7 joints and 6 DOF configuration, the UR10 provides the necessary freedom of movement to reach the target vehicle from any angle while ensuring stability.

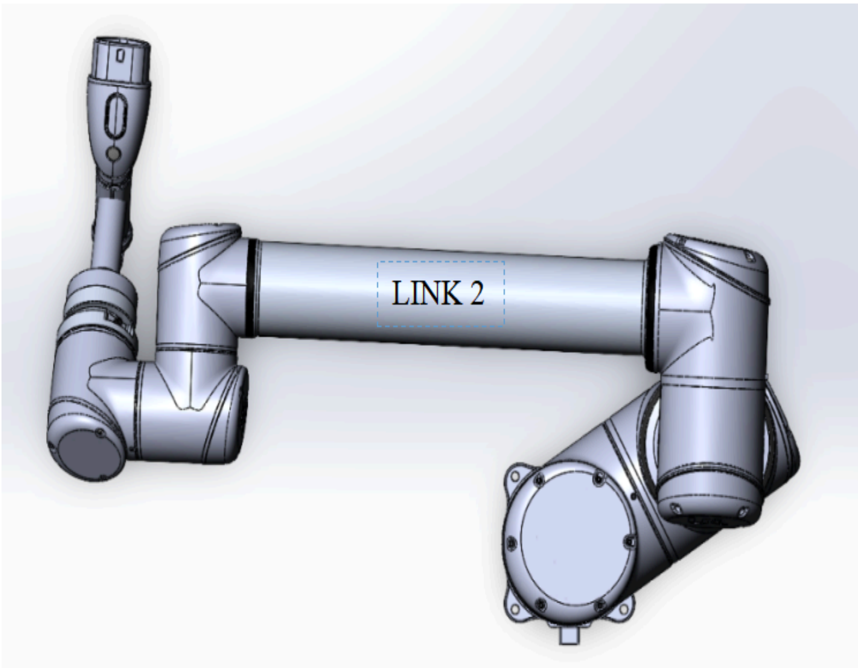
The design incorporates specific dimensions with Link 1 measuring 75.41 cm and Link 2 measuring 70.35 cm, as shown in table 3.1, to achieve the desired reach. The structure is as show in figure 3.1 The manipulator is mounted on an elevation platform with a height of 50 cm, optimizing its functionality. The end-effector has been tailored to accommodate the EV charger plug, aligning with the project's objectives. This design ensures optimal performance in real-world scenarios, meeting both functional requirements and safety standards. The application used to make this design was the 2021 version of SolidWorks. Below Table 3.1 shows the lengths of the links and Figure 3.1(a-d) represents the design model of the same.

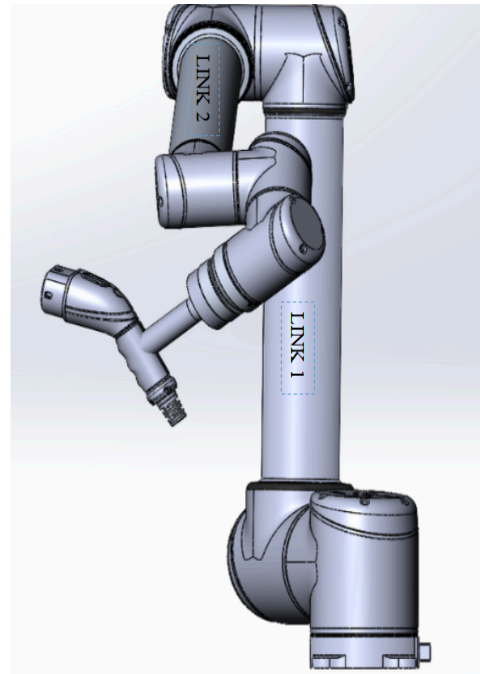
**Table 3.1: Link lengths of robotic arm**

| LINK | LENGTH (cm) |
|---|---|
| Link 1 | 75.41 |
| Link 2 | 70.35 |

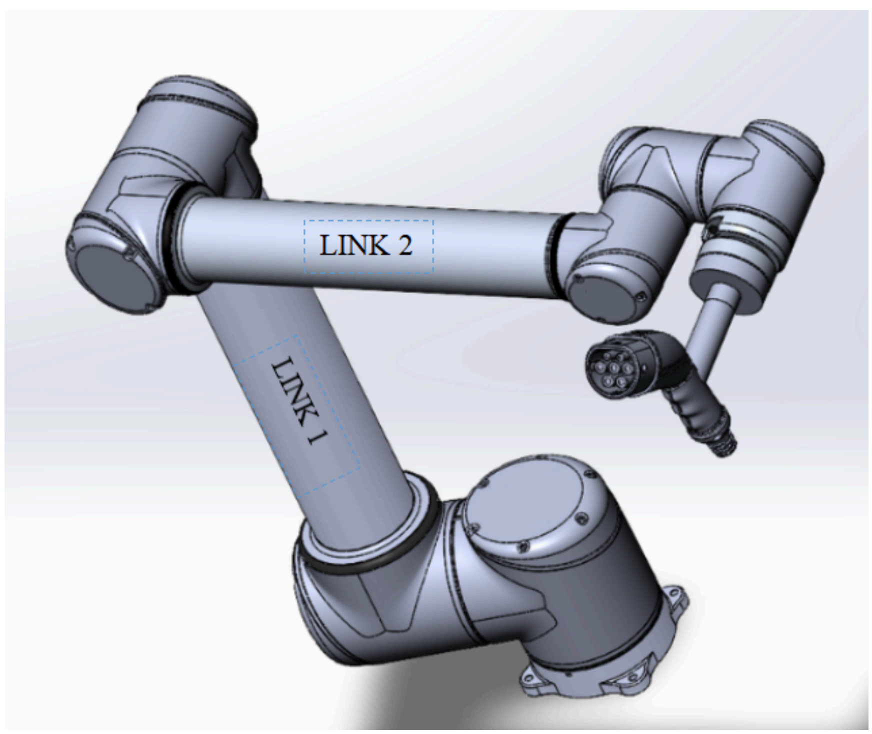



**a)**




**b)** **c)**




**d)**

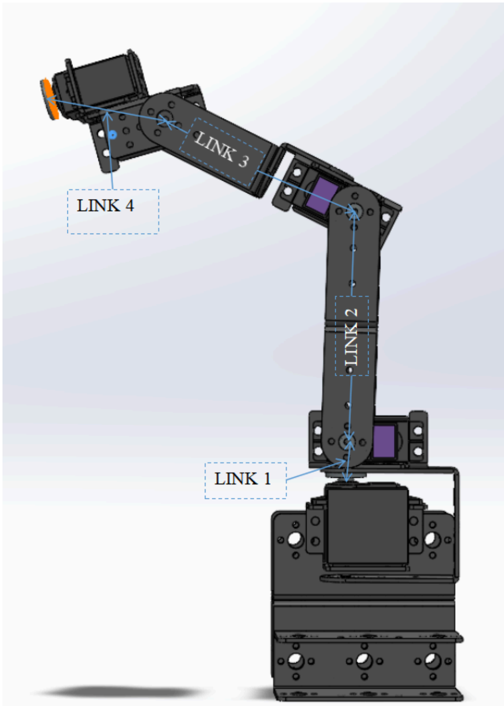
**Figures 3.1: CAD design of robotic arm a) Front view b) Top view c) side view d) diagonal view**

### 3.1.2 Prototype Design

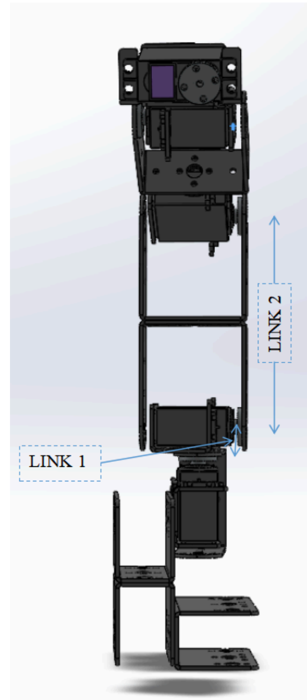
The schematic representations provided herein elucidate the intricate design intricacies of the final product conceptualized for the Major project. With an overarching goal of cost-effectiveness, a deliberate decision was made to fabricate the prototype at a scaled-down ratio of 1:6. Extensive exploration within the repository of manipulator designs on Grabcad led to the discovery of a suitable model that aligned with the stipulated criteria for size and weight capacity. However, the identified model fell short in terms of Degrees of Freedom (DOF), possessing only 5. To overcome this limitation, an innovative solution was devised wherein an additional degree of freedom was attained by integrating the arm onto a horizontally mobile rover platform. The structure is as shown in figure 3.2. The structural integrity of the arm is meticulously crafted from robust metal bracket parts, meticulously calibrated to precise dimensions. Specifically, the lengths of Link1, Link2, Link3, and Link4 are recorded at 1.5 cm, 10.5 cm, 14.7 cm, and 7.3 cm respectively, as shown in table 3.2. The reach of the arm extends to a notable distance of 37.5 cm, rendering it capable of fulfilling its intended functionalities across a substantial operational range. Overall, the design amalgamates ingenuity with practicality culminating in a prototype poised to significantly contribute to the project's objectives.
Table 3.2 represents the new Link lengths.

**Table 3.2: Link lengths of prototype**

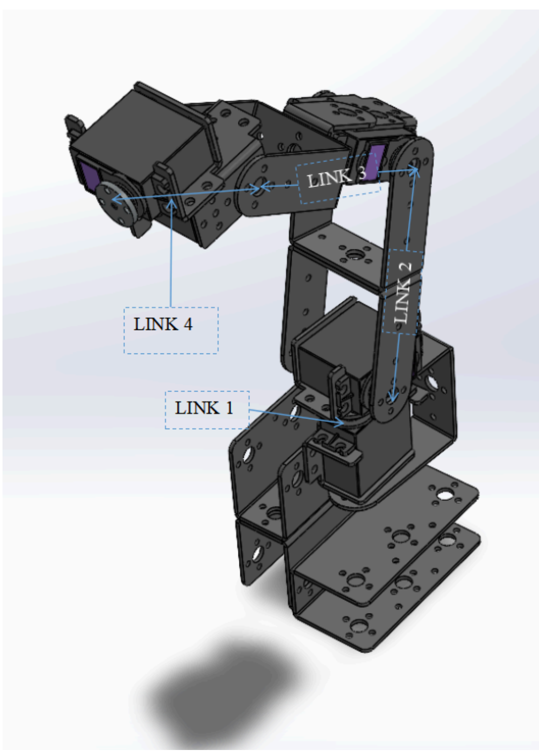
| LINK | LENGTH (cm) |
|---|---|
| Link 1 | 1.5 |
| Link 2 | 10.5 |
| Link 3 | 14.7 |
| Link 4 | 7.3 |




**(a)**




**(b)**




**(c)**

**Figures 3.2: CAD design of Prototype arm**

**a) Side view b) Front view c) Diagonal view**

### 3.1.3 End Effector

**Figure 3.3: CAD design of end effector**

The diagram presented above delineates the intricacies of the end effector designed specifically for the project, serving the pivotal function of a charger. Employing a strategic approach to fabrication, the design is halved along its axis, facilitating easy replication through mirror imaging for efficient 3D printing. Subsequently, the two halves can be seamlessly joined using adhesive techniques. Leveraging advanced CAD software including SolidWorks 2021 and Autodesk inventor, the design was meticulously crafted to ensure precision and functionality. Integration of the end effector onto the arm necessitated precise alignment, achieved through the creation of 1cm diameter holes positioned at equidistant intervals from the arm's terminus. Brass inserts were meticulously embedded into these holes, employing heat to ensure secure fixation, thereby enabling seamless attachment of the end effector to the arm via bolted connections utilizing the brass inserts. Notably, the protruding point intended for insertion into the EV charging port extends 6cm from the arm's terminus, positioned at an angle of 33.557 degrees, meticulously calculated to ensure optimal functionality and compatibility with charging infrastructure. Figure 3.3 shows the CAD-CAM design of end-effector.

### 3.1.4 EV Charging Port Holder Design

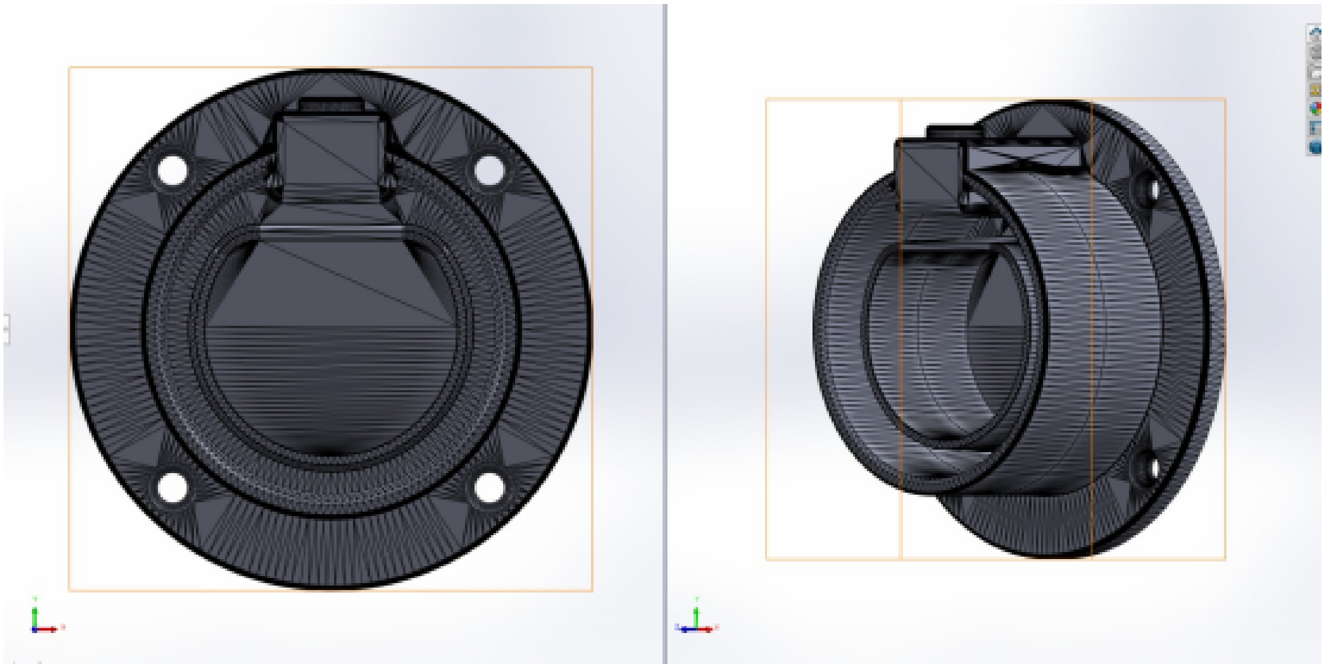


**Figure 3.4: CAD design of EV Charging Port Holder**

The visual depiction provided above showcases the meticulously engineered design of a holder designated to function as the charging port for the Electric Vehicle within our prototype framework. Serving as the docking point for the end effector, this component assumes a critical role in facilitating the charging process. Fabricated utilizing advanced Computer-Aided Design (CAD) software including SolidWorks 2021 and Autodesk Inventor, meticulous attention was dedicated to ensuring precision and compatibility with the intended application. Designed to be affixed onto a vertical platform, the holder is strategically crafted to seamlessly accommodate the insertion of the end effector, fostering a snug and secure fit. This meticulous design approach ensures optimal functionality and interoperability within the EV charging infrastructure, underscoring the commitment to efficiency and reliability in our prototype implementation.

Figure 3.4 shows different views of the Charging Port Holder designed for the application.

## 3.2 SCHEMATIC DIAGRAM

### 3.2.1 Robotic arm circuit

The presented schematic diagram illustrates the circuitry of the prototype developed for the final market product.

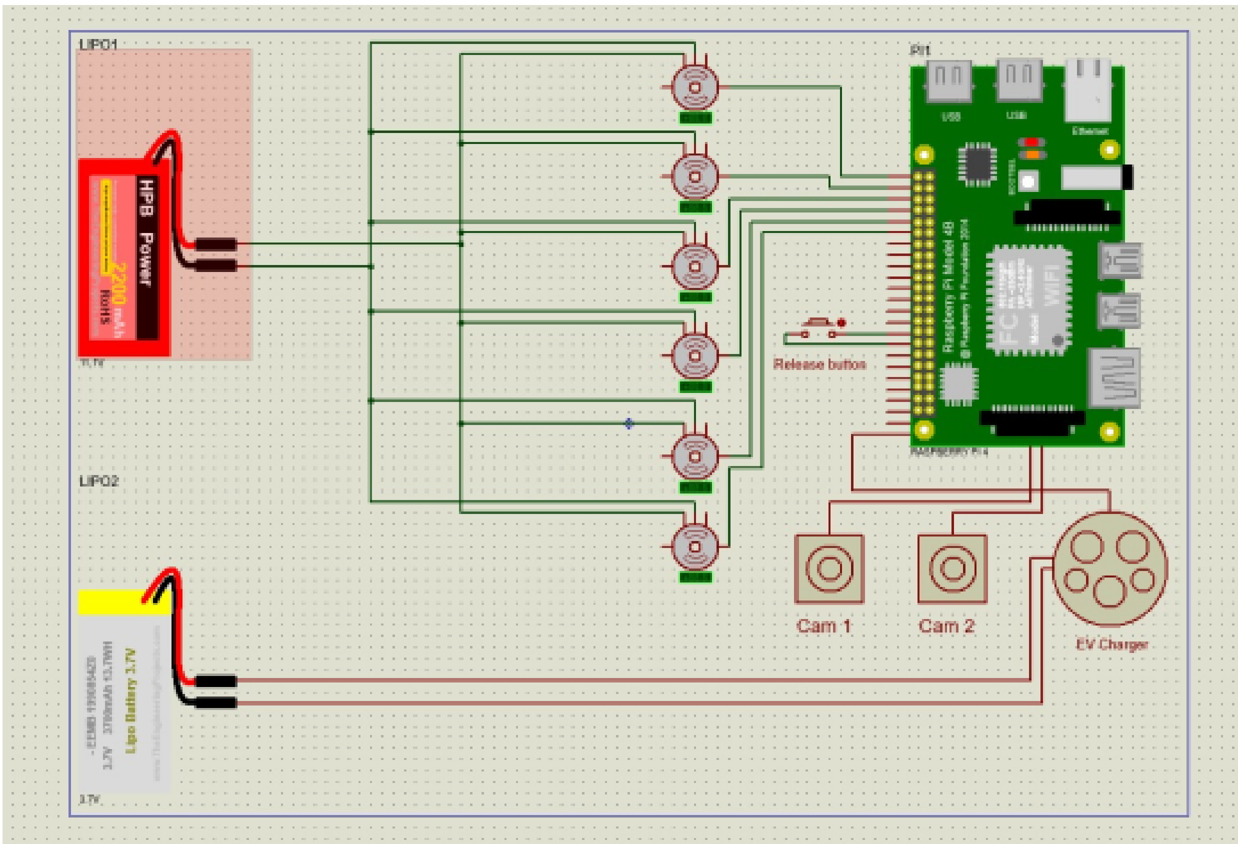



**Figure 3.5: Schematic diagram of robotic arm**

Servo Motors: 6 servo motors are strategically positioned within the robotic arm to facilitate smooth and accurate movement. These servo motors are intricately connected to the Raspberry Pi for seamless integration and precise control over the arm's articulation.

Power Sources: The system incorporates two distinct power sources, each serving specific functionalities. One power source is dedicated to supplying power to the motor driver and microcontroller, ensuring optimal performance and stability. The second power source is designated for the charger, facilitating efficient energy management within the system.

Raspberry Pi 4 (4GB RAM): The central control unit of the entire system, the Raspberry Pi 4 with 4GB of RAM, orchestrates the functionality of all interconnected components. Serving as the nerve center, it facilitates communication and coordination among the various subsystems, ensuring cohesive operation and execution of tasks.

Motor Driver (L298 Module): The L298N motor driver interfaces with the Raspberry Pi to control motors on the four-wheel cart. It regulates motor speed and direction, allowing the cart to move and position the robotic arm as required for charging tasks.

Camera Modules (Cam 1 & Cam 2): Within the system architecture, Camera 1 serves the crucial function of identifying the model of the car, enhancing the system's capabilities for intelligent recognition and decision-making processes. Conversely, Camera 2 is dedicated to pinpointing the precise location of the charging port, optimizing the alignment and interaction between the robotic arm and the charging infrastructure. Both Camera 1 and Camera 2 are seamlessly integrated with the Raspberry Pi, ensuring streamlined communication and data exchange between the cameras and the central control unit.

### 3.2.2 Prototype Circuit

The presented schematic diagram illustrates the circuitry of the prototype developed for the major project showcase.

Motor Driver (L298 Module): The L298N motor driver interfaces with the Raspberry Pi to control motors on the four-wheel cart. It regulates motor speed and direction, allowing the cart to move and position the robotic arm as required for charging tasks.

Servo Motors: Five servo motors are strategically positioned within the robotic arm to facilitate smooth and accurate movement. These servo motors are intricately connected to the Raspberry Pi for seamless integration and precise control over the arm's articulation.

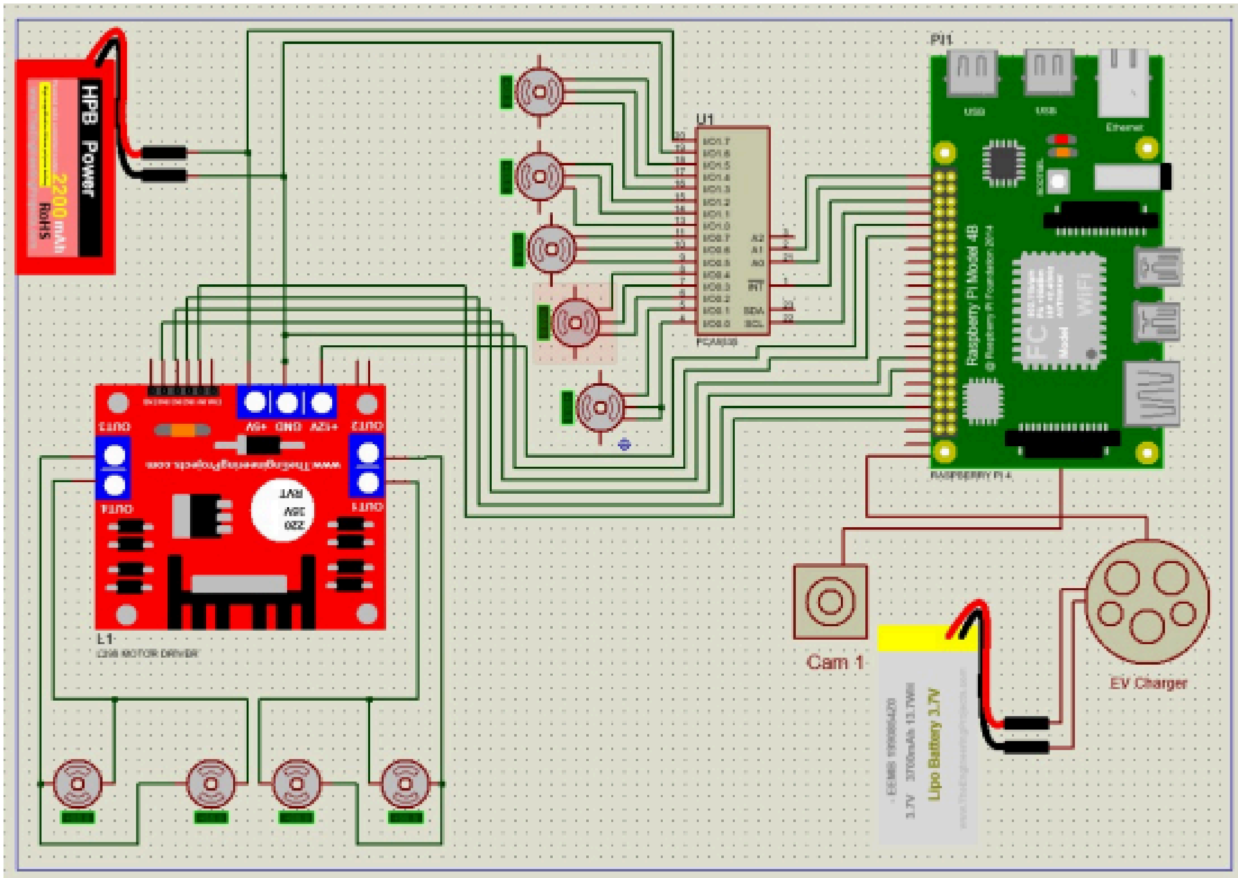



**Figure 3.6: Schematic diagram of prototype**

PCA9685 Driver: The PCA9685 driver interfaces with the Raspberry Pi to control servo motors on the robotic arm. It generates PWM signals to adjust servo motor positions, facilitating precise arm movements.

DC Motors: Four 150rpm Dual Shaft BO Motors are integrated into the prototype, specifically tailored for DIY projects, notably robotics and smart car applications. These motors are interconnected with the motor drive system to ensure synchronized operation and precise control.

Power Sources: The system incorporates two distinct power sources, each serving specific functionalities. One power source is dedicated to supplying power to the motor driver and microcontroller, ensuring optimal performance and stability. The second power source is designated for the charger, facilitating efficient energy management within the system.

Raspberry Pi 4 (4GB RAM): The central control unit of the entire system, the Raspberry Pi 4 with 4GB of RAM, orchestrates the functionality of all interconnected components. Serving as the nerve center, it facilitates communication and coordination among the various subsystems, ensuring cohesive operation and execution of tasks.

Camera Module (Cam 1): A camera module, denoted as Cam 1, is integrated into the system for dual purposes. Firstly, it enables the detection of the model of the car, enhancing the system's capabilities for intelligent decision-making. Additionally, it facilitates the precise detection of the position of the port connected to the Raspberry Pi, ensuring accurate alignment and interaction with external components.

### 3.3 PHYSICAL PROTOTYPE

a) b)

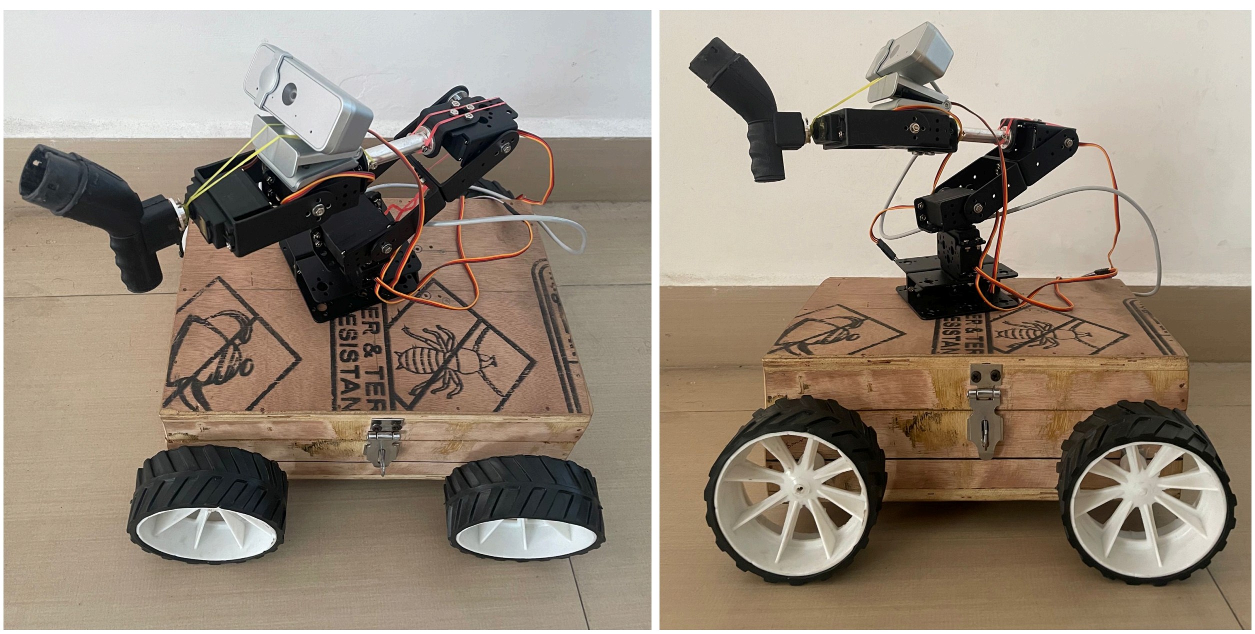


**Figure 3.7: working prototype a) Top view b) Side view**

## 3.4 COMPONENTS SPECIFICATIONS

### 995 DC Motor

The 995 motor is a commonly utilized DC motor in the realm of robotics and DIY projects. Operating on direct current, it typically operates within a voltage range of 3V to 12V DC, making it adaptable to various battery setups commonly employed in hobbyist pursuits. What distinguishes the 995 motor is its impressive torque output relative to its size, rendering it well-suited for tasks requiring substantial mechanical power, such as driving wheels or lifting mechanisms in robotics endeavors. Featuring a built-in gearbox or gear reduction system, the 995 motor efficiently transforms its rotational motion, resulting in amplified torque and reduced speed at the output shaft. This geared setup, often with ratios like 1:48 or 1:120, enhances its effectiveness in activities necessitating precise management of speed and force. Despite its robust performance, the 995 motor maintains a compact design, ensuring compatibility with projects constrained by space limitations. With its widespread availability and cost-effectiveness, the 995 motor remains a favored option among hobbyists and enthusiasts for a wide range of applications, spanning from small-scale robotics to intricate automation projects.


**Figure 3.8: MG995 Servo Motor**

### Raspberry Pi 4 (4GB RAM)

The Raspberry Pi 4, which boasts 4GB of RAM, is a compact and powerful single-board computer that is highly regarded for its versatility and affordability. Featuring a quad-core ARM

Cortex-A72 processor that can run at speeds of up to 1.5GHz, it offers improved performance for handling multiple tasks and demanding computational workloads. Its wide range of connectivity options, including dual micro-HDMI ports, USB 3.0 and 2.0 ports, Gigabit Ethernet, and dual-band 802.11ac wireless capabilities, ensures seamless integration with various displays, peripherals, and networks. The Raspberry Pi 4 remains compatible with a vast selection of accessories and software, making it an excellent platform for experimentation, prototyping, and educational purposes in diverse fields.

When used as the central processing unit for controlling a 5-degree-of-freedom (5DOF) robotic arm, the Raspberry Pi leverages its computing power and GPIO capabilities effectively. By connecting the Raspberry Pi to motor driver modules or controllers, users can precisely control the movement of each joint of the robotic arm. Through programming in languages like Python, the Raspberry Pi can execute commands to adjust the position and speed of the motors, enabling the robotic arm to perform tasks such as object manipulation, assembly operations, or complex movements in educational settings. Additionally, the Raspberry Pi's connectivity features enable remote control and monitoring of the robotic arm, making it a versatile and accessible platform for robotics enthusiasts, hobbyists, and educational applications.

**Figure 3.9: RaspberryPi 4**

### Geared 150-RPM Motor

The Robotbanao 150RPM Dual Shaft BO Motor is a motor designed for use in DIY projects, specifically for robotics and smart car applications. Its dual shaft design and L-shaped configuration allow for flexible mounting and integration into various projects. The motor is geared to operate at a speed of 150 revolutions per minute (RPM), making it appropriate for tasks requiring moderate rotational speed and torque. The motor's yellow color makes it visually distinct and can help in identifying and coordinating components within a project. This motor is designed to provide dependable performance and ease of use for hobbyists and enthusiasts creating their own robotic devices.

**Figure 3.10: DC Motor (12V)**

### OT5320M Copper Metal Gear Digital Servo Motor

The Orange OT5320M is a high-performance digital servo motor that operates on a 7.4V power supply. It is capable of delivering a torque output of 20kg.cm, making it a powerful and versatile servo motor.

The motor has a rotation range of approximately 120 degrees, with 60 degrees of movement in each direction. This wide range of motion allows for precise control and positioning of the attached components.

The servo motor features a copper metal gear design, which provides enhanced durability and strength compared to standard plastic gears. This makes the OT5320M suitable for applications that require reliable and long-lasting performance.

The motor package includes a selection of hardware and arms, providing users with the necessary accessories to integrate the servo into their projects. The wire connections are clearly labeled, with the red wire indicating the power supply, the black wire for ground, and the white wire for the control signal.

Overall, the Orange OT5320M 7.4V 20kg.cm 180° Copper Metal Gear Digital Servo Motor is a high-quality and robust servo motor that can be used in a variety of applications, such as robotics, model making, and other projects that require precise and powerful motion control.

**Figure 3.11: OT5320M Copper Metal Gear Digital Servo Motor**

**PCA9685 Driver**

The PCA9685, an IC driver by NXP Semiconductors, facilitates precise PWM signal generation for up to 16 servos or LEDs. Operating via I2C, it communicates with microcontrollers, boasting adjustable frequency and supporting multiple devices on the same bus. Widely employed in robotics, lighting systems, and motion control applications, its features include 12-bit resolution, hardware reset capability, and an internal oscillator, enabling tailored control over various actuators and lighting elements.

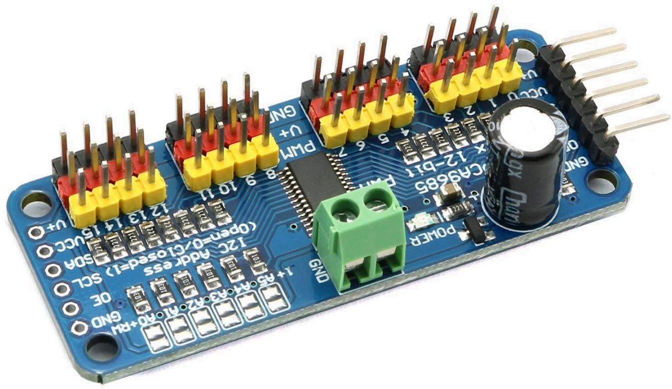


**Figure 3.12: PCA9685 Driver**

### L298N Motor Driver

The L298N is an integrated circuit serving as a motor driver, popular for its ability to control DC or stepper motors bidirectionally. It facilitates independent regulation of motor speed and direction for two motors. Accepting TTL logic signals for control, it employs an H-bridge configuration, ensuring efficient motor management. Widely utilized in robotics and automation, the L298N is prized for its versatility and reliability in driving various motor types in diverse applications.

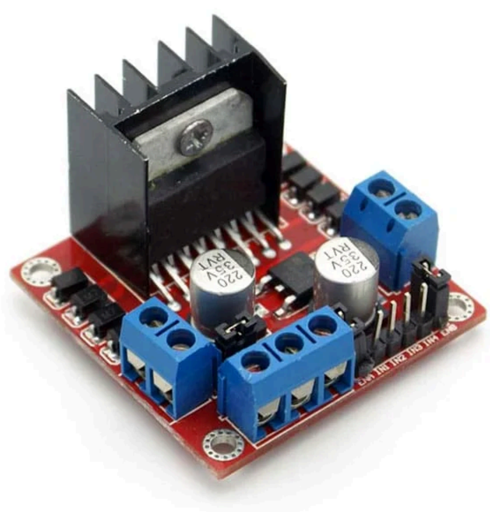


**Figure 3.13: L298N Motor Driver**

# CHAPTER-4

# METHODOLOGIES AND SIMULATION

## 4.1 MACHINE LEARNING

### 4.1.1 Machine Learning

Machine learning, a subset of artificial intelligence, enables systems to learn from data and improve their performance without being explicitly programmed. In the context of detecting car models from camera input, machine learning employs a supervised learning approach, where the model learns to make predictions based on labeled training data.

1. Data Collection: The process begins by gathering a dataset comprising images of various car models, labeled with their corresponding model names. It's essential to include diverse examples capturing different angles, lighting conditions, and backgrounds to ensure the model's robustness.

2. Preprocessing: Following data collection, preprocessing steps are applied to enhance image quality and standardize them for training. This includes resizing images, applying color normalization, and removing noise or artifacts. Preprocessing helps in improving the quality of input data and facilitates better learning by the model.

3. Feature Extraction: The model extracts relevant features from input images to distinguish between different car models. Convolutional Neural Networks (CNNs) are commonly utilized for this task due to their ability to learn hierarchical representations of features from raw pixel data. CNNs automatically learn patterns and features from images, making them suitable for tasks like image classification.

4. Model Training: Using the preprocessed data and extracted features, the model undergoes training using a supervised learning algorithm, such as a CNN. During training, the model learns to map input images to their corresponding car model labels by adjusting its parameters to minimize the difference between predicted and true labels. Training involves iteratively

presenting the model with labeled examples and updating its internal parameters based on the errors made in predictions.

5. Evaluation: Once trained, the model's performance is evaluated using a separate validation dataset to assess its accuracy, precision, recall, and other relevant metrics. Evaluation helps in gauging how well the model generalizes to unseen data and ensures its reliability in real-world scenarios.

6. Deployment: Once the model achieves satisfactory performance, it can be deployed in real-world applications to detect car models from camera input. This may involve integrating the model into a software application or system that processes live camera feeds and identifies car models in real-time.

By leveraging machine learning techniques, particularly supervised learning with CNNs, accurate and robust models capable of detecting car models from camera input can be developed. This enables various applications such as vehicle identification, security surveillance, and automotive analytics.

### 4.1.2 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are pivotal in automatically discerning features from images, critical for tasks like car model detection from camera input and identification of electric vehicle (EV) charging ports. In car model detection, CNNs extract distinguishing features from images, categorizing them into specific models. For charging port detection, CNNs utilize object detection methods to pinpoint and classify ports within images. Fine-tuning pretrained CNN models expedites training and enhances performance, leveraging previously learned features for better adaptation to new tasks. Harnessing CNNs streamlines and enhances recognition accuracy, advancing EV integration into modern transportation infrastructure. These applications play a vital role in shaping smart charging systems, autonomous driving technologies, and sustainable transportation networks. With continued advancements in CNNs and machine learning, the capabilities of EVs in our evolving mobility landscape are poised to further accelerate, promoting cleaner and more efficient transportation solutions.

### 4.1.3 Teachable Machine

Teachable Machine is a web-based tool developed by Google that enables users to create machine learning models without requiring extensive coding knowledge. It allows users to train custom machine learning models using their webcam and microphone, making it accessible for beginners and experts alike.

Here's how Teachable Machine works:

1. Data Collection: Users begin by collecting data through their webcam or microphone. They can capture images, audio, or pose data to train their machine learning model. For example, users might collect images of different objects they want to classify or record audio samples of different sounds they want to recognize.

2. Training: Once the data is collected, users label the data to indicate the desired outputs. For instance, they might label images of cats and dogs or classify audio samples as "yes" or "no." Teachable Machine then uses this labeled data to train a machine learning model behind the scenes.

3. Testing and Evaluation: After training, users can test their model in real-time using the webcam or microphone. Teachable Machine provides immediate feedback on how well the model performs, allowing users to refine their training data or model parameters as needed.

4. Exporting: Once satisfied with the model's performance, users can export it for use in their own projects. Teachable Machine provides options to export the model as TensorFlow.js code for use in web applications, or as a TensorFlow Lite model for use in mobile applications.

Overall, Teachable Machine democratizes machine learning by providing a user-friendly interface for creating custom models. It empowers users to explore the possibilities of machine learning without the need for extensive programming knowledge, opening up opportunities for creative experimentation and innovation in various domains.

### 4.1.4 Google Colab

Google Colab is a cloud-based platform for writing and running Python code collaboratively. It allows users to create and share Jupyter notebooks containing code, text, and visualizations. Through shared notebooks, users can easily share their code with others, fostering collaboration and knowledge sharing within the Python programming community.

Colab enables users to access code from other users through various means. Firstly, shared notebooks on Colab are often made publicly accessible, allowing users to discover and access code written by others on a wide range of topics. Additionally, Colab seamlessly integrates with popular Python libraries and frameworks, enabling users to import code from these libraries into their own notebooks. Furthermore, Colab supports importing notebooks directly from GitHub repositories, streamlining the process of accessing and using code shared by other users and developers.

Overall, Google Colab provides an accessible and collaborative platform for sharing and using Python code, empowering users to leverage the collective knowledge and expertise of the community to accelerate their own learning and development efforts.

### 4.1.5 LabelImg

LabelImg is an open-source graphical image annotation tool used for labeling object bounding boxes in images. It is commonly used in computer vision projects, particularly in the development of datasets for training object detection algorithms. It provides a user-friendly interface for annotating images by drawing bounding boxes around objects of interest and assigning corresponding labels to each box. These annotations serve as ground truth data for training machine learning models to detect and classify objects within images accurately. It supports various annotation formats, including Pascal VOC XML and YOLO TXT, making it compatible with popular deep learning frameworks such as TensorFlow, PyTorch, and Darknet. With its simplicity and versatility, LabelImg is widely used by researchers, developers, and data scientists in the field of computer vision to streamline the process of creating labeled datasets for training and evaluating object detection models.

## 4.2 TRAINING OF MACHINE LEARNING MODEL

### 4.2.1 Iteration #1 : Using LabelImg and Google Colab

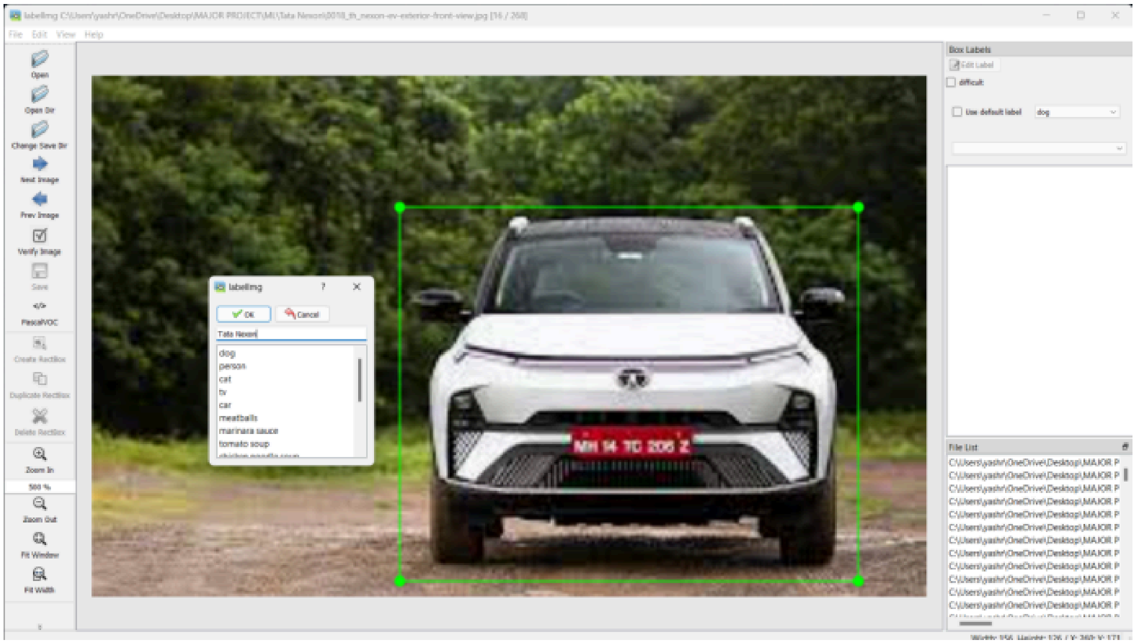
1. Objective Definition: The project aimed to identify cars in India that feature side charging ports, excluding those on the front or rear. This task required careful examination of several car models to determine their charging port locations, ensuring the accuracy of the data collected.

2. Car Selection: The project selected several car models commonly found in India, including the Tata Nexon, Tata Nano, Mahindra XUV 400, Tata Tiago, and Hyundai Ioniq. These models represent a diverse range of vehicles, covering different segments and manufacturers prevalent in the Indian automotive market.

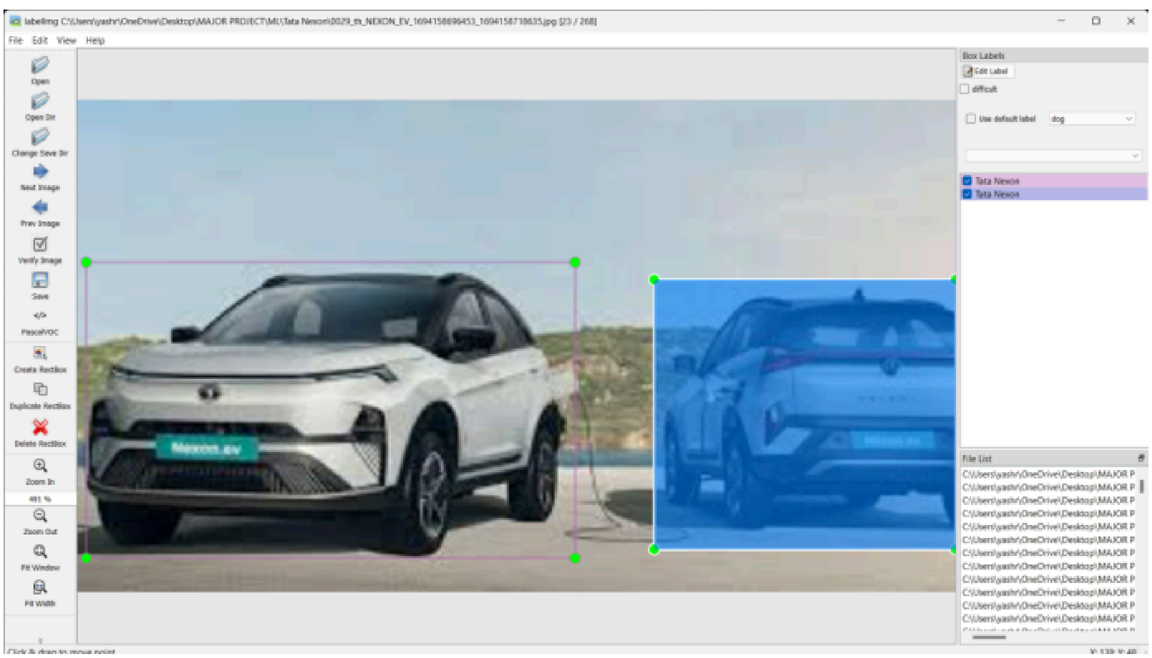
3. Data Collection: To begin the project, a comprehensive dataset of images featuring the selected car models was compiled. This involved downloading over 200 high-quality images of each car, showcasing various colors and backgrounds.

4. Data Filtering: Following data collection, a rigorous filtering process was implemented to remove low-quality images and those showing car interiors. This step was crucial to maintain the integrity of the dataset and ensure that only relevant images depicting exterior views of the cars were included for analysis.

5. Annotation Process: The filtered images were then annotated using the LabelImg tool. Annotation involved drawing bounding boxes around the side charging ports of the cars and assigning corresponding labels. This process provided ground truth data for training machine learning models to accurately identify the charging port locations in the images.




**(a)**




**(b)**

**Figures 4.1(a,b) : The images display how the car, Tata Nexon, was annotated using LabelImg.**

6. Quality Control: After annotation, the annotated images were carefully reviewed to identify any discrepancies or inaccuracies in the annotations. This step was essential to ensure the accuracy and consistency of the annotated dataset, as any errors could impact the performance of the machine learning models trained on the data.

7. Machine Learning Setup: With the annotated dataset prepared, a Google Colab notebook was utilized to set up machine learning for car identification. This involved importing various repositories from GitHub and configuring the environment for machine learning training.

8. Data Preparation: The annotated images were divided into train, validation, and test sets in an 8:1:1 ratio. This partitioning ensured that the machine learning model could be trained on a diverse range of data while also evaluating its performance on unseen data. Table 4.1 shows the number of images used for the training, validation and testing in the dataset.

| Car | Training (80%) | Validation (10%) | Testing (10%) | Total |
|---|---|---|---|---|
| Hyundai Ioniq | 267 | 33 | 33 | 333 |
| Tata nano | 304 | 25 | 25 | 254 |
| Tata Nexon | 201 | 25 | 25 | 251 |
| Tata Tiago | 151 | 19 | 19 | 189 |
| XUV 400 | 134 | 16 | 16 | 166 |

**Table 4.1: This data shows the total number of images of each car used for training, validation and testing purposes.**

9. Model Selection: For machine learning training, the SSD-MobileNet-V2 model was chosen due to its efficiency and effectiveness in object detection tasks. Its lightweight and efficient design make it a popular choice for real-time object detection tasks where both speed and accuracy are essential. With ongoing advancements in neural network architectures and

optimization techniques, SSD-MobileNet-V2 continues to push the boundaries of what is possible in the field of object detection. The model architecture was well-suited for identifying charging ports in images while maintaining high performance. Below Figure 4.2 represents the architecture of SSD-Mobilenet-V2.

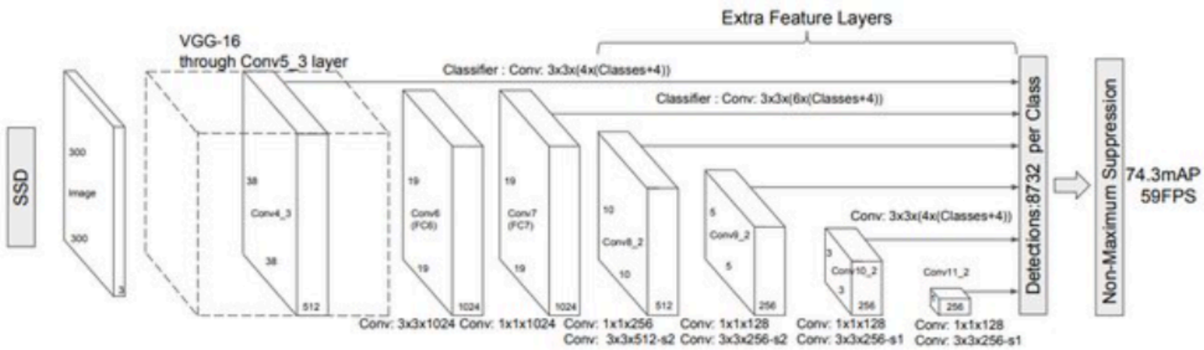



**Figure 4.2: Architecture of SSD-Mobilenet-V2**

| Model | Inference Time (ms) | Throughput (FPS) | Throughput w/ Edge TPU (FPS) |
|---|---|---|---|
| SSD-MobileNet-v2 (FP32) | 142.1 | 2.85 | N/A |
| SSD-MobileNet-v2 (INT8) | 68.96 | 3.83 | 21.8 |
| SSD-MobileNet-v2-FPNLite (FP32) | 169.2 | 2.23 | N/A |
| SSD-MobileNet-v2-FPNLite (INT8) | 95.08 | 2.83 | 8.49 |
| SSD-MobileNet-v1 (INT8) | 65.59 | 3.76 | 22.1 |
| EfficientDet-Lite-D0 (INT8) | 112.6 | 2.58 | 13.0 |
| EfficentDet-D0 (FP32) | 1520 | 0.55 | N/A |

**Table 4.2: Comparison of available CNN architectures**

Table 4.2 reflects the comparison of available CNN architectures and Figure 4.3 is the Graphical representation of the same.

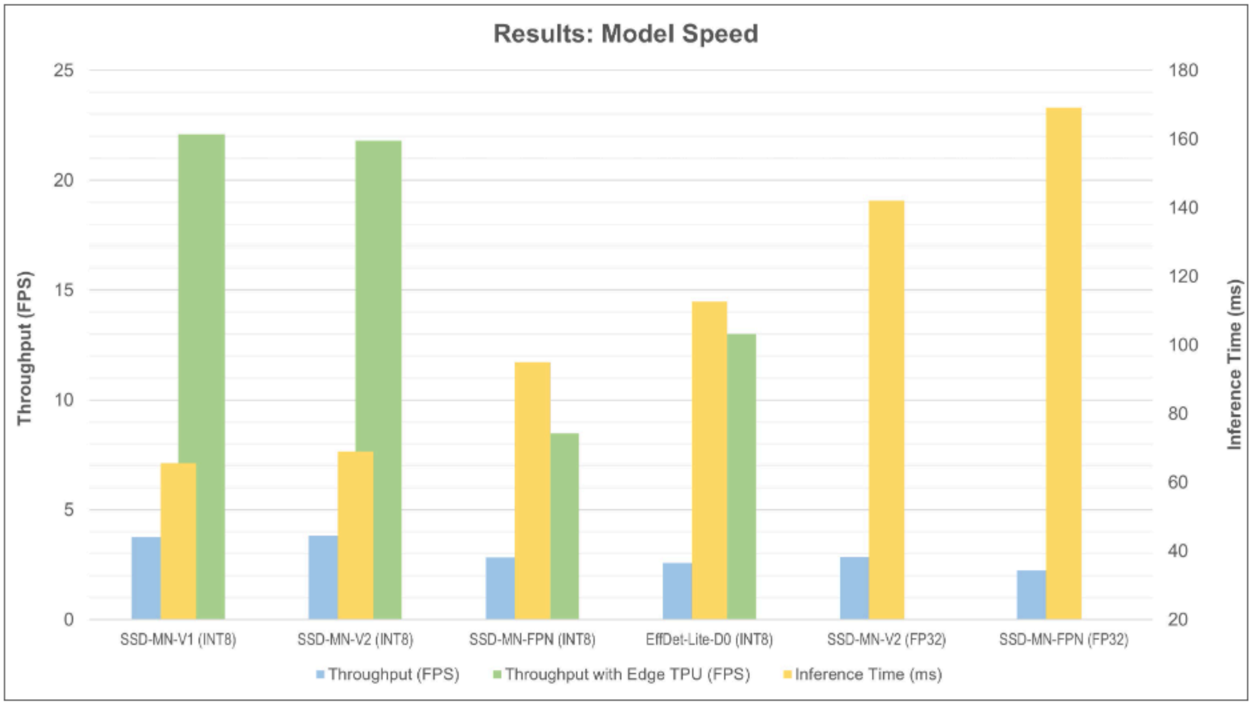



**Figure 4.3: Graphical comparison of CNN architectures**

| Model | Accuracy Score (COCO mAP @ 0.5:0.95) | Objects Correctly Labeled (out of 335) |
|---|---|---|
| SSD-MobileNet-v2 (FP32) | 75.33% | 306 |
| SSD-MobileNet-v2 (INT8) | 60.99% | 264 |
| SSD-MobileNet-v2-FPNLite (FP32) | 84.81% | 326 |
| SSD-MobileNet-v2-FPNLite (INT8) | 76.41% | 302 |
| SSD-MobileNet-v1 (INT8) | 49.77% | 225 |
| EfficientDet-Lite-D0 (INT8) | 60.79% | 302 |
| EfficentDet-D0 (FP32) | 46.91% | 235 |

**Table 4.3: Comparison based on Accuracy of CNN Architectures**

It can be inferred from Table 4.2 and Table 4.3 that SSD-mobilenet-V2 provides maximum accuracy and throughput(FPS) and thus is optimal for usage in current application.

10. Training Parameters: The machine learning model was trained using a total of 40,000 steps, with a batch size of 16. Five total classes were created based on the selected car models, allowing the model to distinguish between different charging port locations.

11. Loss Monitoring: To monitor the training progress, TensorBoard was set up to track the loss function. This provided insights into the model's performance over time and helped identify any issues or anomalies during training.

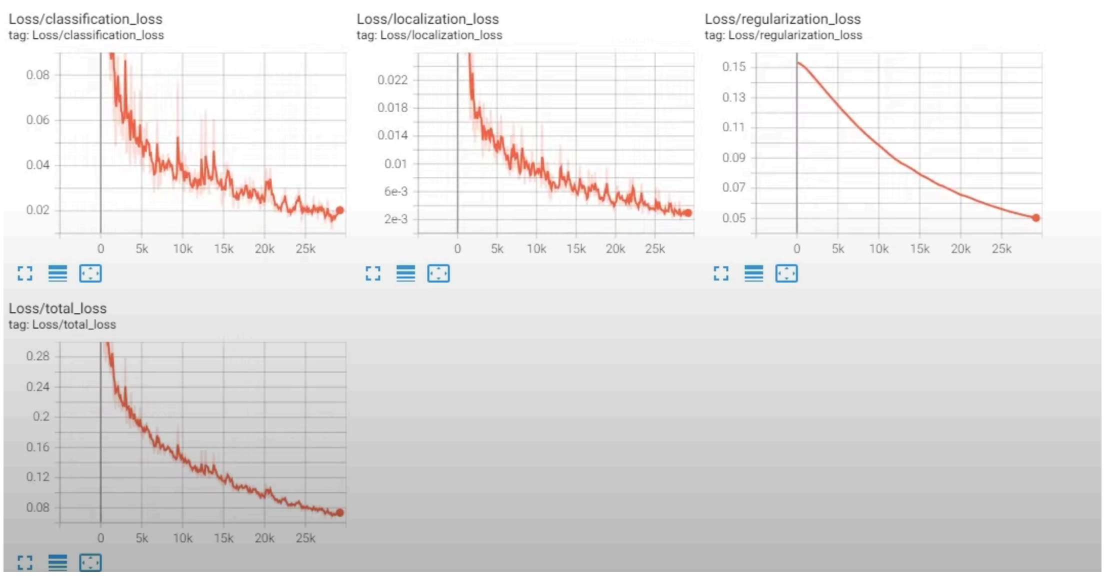



**Figure 4.4: (a)Classification Loss; (b)Localization Loss; (c)Regularization Loss; (d)Total Loss**

As seen from Figure 4.4, the Loss keeps on decreasing as the training continues. The training of Machine Learning can be stopped once losses are stable.

12. Training Execution: The training process commenced, with each step taking approximately 0.03 seconds to complete. Regular updates were provided after every 100 steps to track the training progress and ensure that the model was converging towards optimal performance.

13. Challenges Faced: Despite the initial success of the training process, challenges arose when GPU unit limitations were encountered after 25,000 steps. This setback resulted in the loss of all training data, necessitating a restart from step 6.

14. Outcome and Conclusion: Despite multiple attempts, satisfactory results were not achieved due to the encountered challenges. As a result, the methodology was discarded, and alternative approaches were explored to address the limitations and improve the project's outcomes in future iterations.

In summary, the project involved a systematic process of data collection, annotation, machine learning setup, and training to identify cars in India with side charging ports. While the initial steps were executed successfully, challenges encountered during training led to the abandonment of the methodology. However, the project provided valuable insights into the complexities of machine learning training and highlighted areas for improvement in future iterations.

**4.2.2 Iteration #2 : Using Teachable Machine**

1. Access Teachable Machine: The Teachable Machine can be accessed through the following link. As discussed above, this can be used to train a Machine Learning model online without going through the process of setting up the whole environment, parameters and directly using a dataset of annotated pictures.

2. Choose Project Type: The Image Project option is chosen to create a project for image classification.

3. Collect Data: Clicking on the "Gather" button, begin the process of collecting data for the project. Annotated images of the five car models are utilized: Tata Nexon, Tata Nano, Mahindra XUV 400, Tata Tiago, and Hyundai Ioniq. Ensuring a diverse set of examples for each car model is aimed to improve the model's accuracy.

4. Train the Model: Teachable Machine automatically segregated the data into training and validation sets, employing transfer learning techniques to train the model effectively. The parameters for the learning process can be altered according to need. Table 4.4 shows the parameters used.

| Parmeters | Epoch | Batch Size | Learning Rate |
|---|---|---|---|
| Values | 50 | 16 | 0.001 |

**Table 4.4: Training Parameters**

5. Evaluate Performance: Upon completion of training, the model's performance is evaluated using the validation set. By analyzing metrics such as accuracy and loss, we assessed how well the model generalized to new data. Below Table 4.5 presents the images used to test the ML model and accuracy of the model in each iteration. This is used to calculate the average accuracy of the model.

| Iteration No. | Sample Space | Correct Predictions | Incorrect Predictions | Accuracy (in %age) |
|---|---|---|---|---|
| 1 | 10 | 10 | 0 | 100 |
| 2 | 10 | 9 | 1 | 90 |
| 3 | 10 | 8 | 2 | 80 |
| 4 | 10 | 10 | 0 | 100 |
| 5 | 10 | 9 | 1 | 90 |

**Table 4.5: Performance of ML Model**

Average Accuracy Percentage of Machine Learning Model = 92%

6. Test the Model: Utilizing the "Test" tab, the model is tested with new input data. Using both the camera and uploaded images, predictions are made based on the training and observed real-time feedback on its performance.

7. Export the Model: Upon satisfaction with the model's performance, export it for use in the project. Opting for TensorFlow Lite and TensorFlow.js formats ensure compatibility with various applications and platforms.

8. Integrate the Model: Finally, integrate the exported model into the project, leveraging its machine learning capabilities for image classification of the five car models. Whether for real-time classification or batch inference, the model is utilized to enhance the project's functionality and performance.

By following these detailed steps, Teachable Machine can be effectively utilized for training a machine learning model to classify images of the five car models, ensuring accuracy and reliability.

**4.3 SOFTWARE**

**4.3.1 Robot Operating System (ROS) Integration and Transition to Python**

ROS (Robot Operating System): ROS is an open-source framework designed for robotic systems development. It provides a collection of tools, libraries, and conventions aimed at simplifying the creation of complex and robust robot software. ROS facilitates tasks such as hardware abstraction, communication between components, package management, and visualization.

Problems Encountered Using ROS: While ROS offers powerful capabilities for robotics projects, challenges arose during the development of Electrohub:

1. Hardware Limitations: The ROS2 framework demands substantial computational resources, leading to performance issues on the development machine, particularly during the compilation of MoveIt2 for robotic arm control.
2. Complex Setup: Setting up and configuring ROS2, along with dependencies like MoveIt2, required extensive system resources and often resulted in compatibility issues.

Alternative Approach - Transition to Python: To address these challenges, Electrohub transitioned to Python for software development. Python's simplicity, extensive libraries, and compatibility with Raspberry Pi offered a more efficient and manageable solution for controlling hardware components and implementing algorithms.

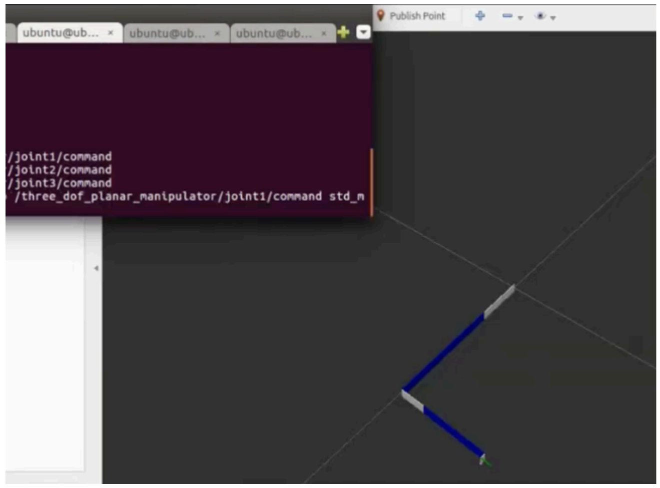



**Figure 4.5: Simulated robotic arm in rviz**

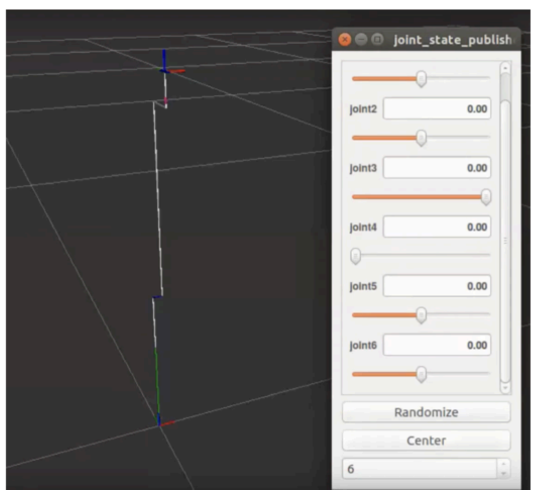



**Figure 4.6: GUI control for joint_state_publisher**

Python: Python is a high-level programming language known for its readability, versatility, and extensive libraries. It is widely used in robotics for its ease of integration with hardware interfaces, mathematical computations, and machine learning frameworks.

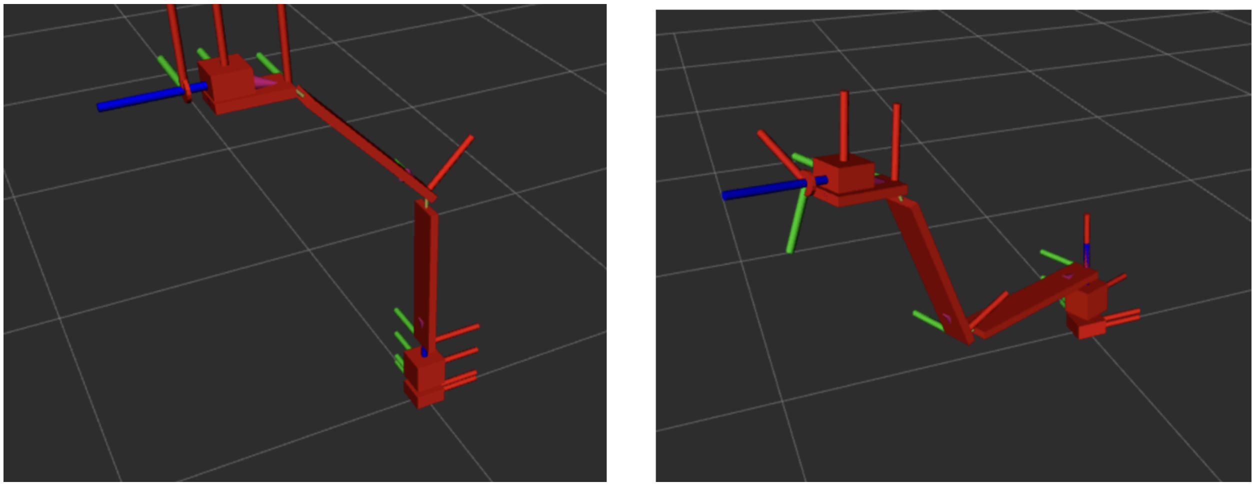


**Figure 4.7 : URDF visualization of robotic arm**

OpenCV (Open Source Computer Vision Library): OpenCV is a powerful library for computer vision tasks in Python. It provides functions and algorithms for image processing, object detection, feature extraction, and more. OpenCV's capabilities were leveraged in Electrohub for tasks such as distance calculations, pose estimation, and orientation detection using image data from the camera mounted on the robotic arm's end effector.

### 4.4.2 Kinematics Calculations and Control

Kinematics: Kinematics is a branch of mechanics that deals with the motion of objects without considering the forces causing the motion. In the context of robotics, kinematics refers to the study of how robotic systems move and position themselves in space.

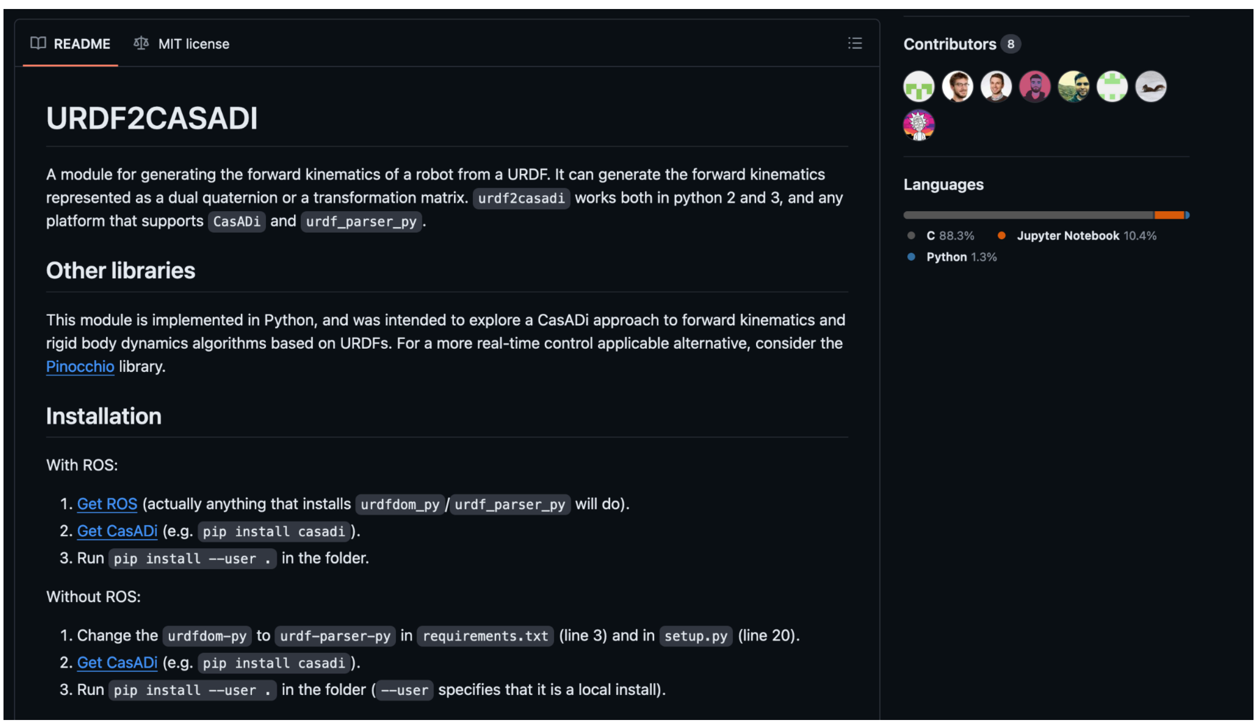
Inverse Kinematics (IK): Inverse kinematics involves determining the joint configurations of a robotic system that result in a desired end effector position and orientation. It's crucial for precise control of robotic arms and manipulators.

Forward Kinematics (FK): Forward kinematics calculates the position and orientation of the end effector based on the joint angles and link lengths of the robotic arm. It helps in understanding how the arm moves in response to given joint configurations.

Python Libraries for Kinematics: Electrohub utilized the urdf2casadi Python library for kinematics calculations. This library integrates with URDF (Unified Robot Description Format) files to derive symbolic expressions for inverse and forward kinematics.

Workflow with urdf2casadi:

1. URDF Utilization: The URDF file of the robotic arm provided the necessary geometric and kinematic information about the arm's structure, joint types, and link lengths.
2. Symbolic Kinematics Derivation: urdf2casadi transformed the URDF data into symbolic expressions for inverse and forward kinematics, allowing precise mathematical calculations.
3. Control Implementation: The derived kinematics equations were used to control the robotic arm's end effector, enabling accurate positioning and orientation required for autonomous charging operations.



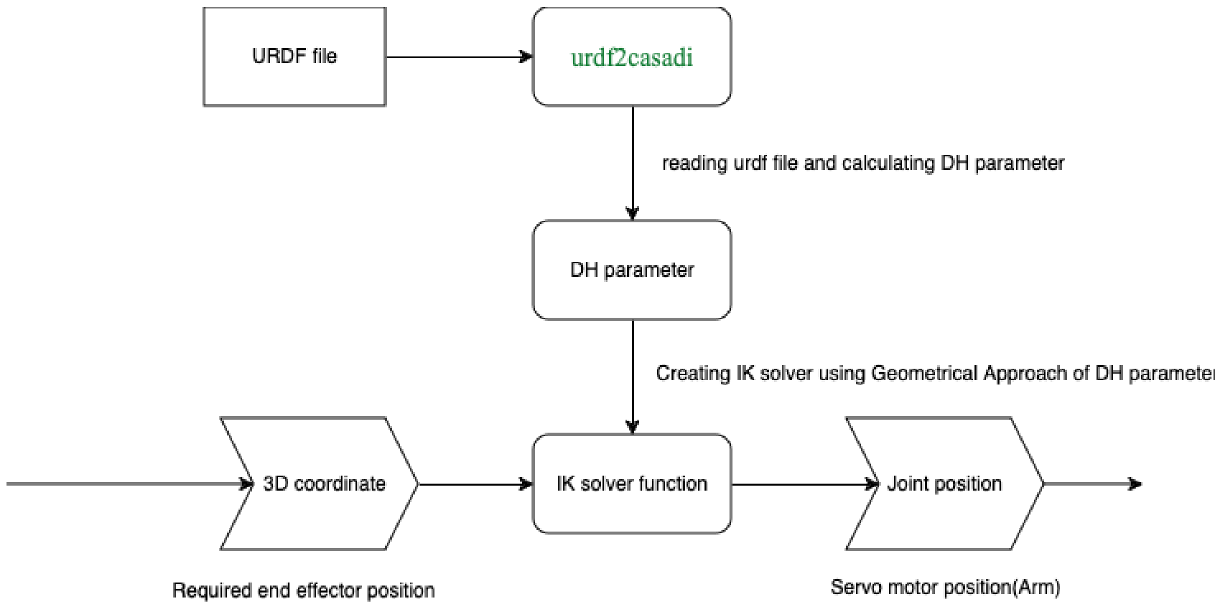



**Figure 4.8: Flowchart of working of urdf2casadi**

By leveraging Python libraries like urdf2casadi and OpenCV, Electrohub achieved efficient kinematics calculations, control, and computer vision capabilities essential for its autonomous charger plugin system for electric vehicles.

### 4.3.3 Camera Calibration

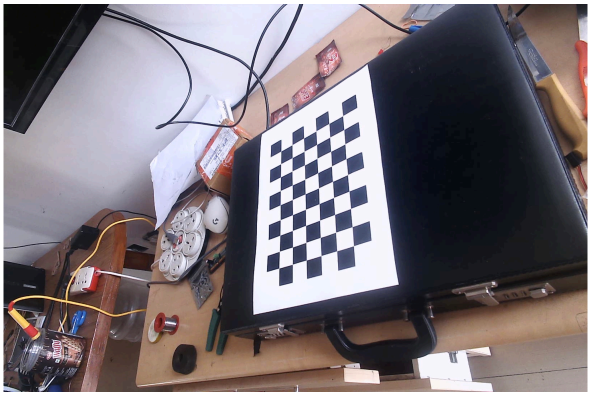
Camera Calibration Process: Camera calibration is a critical step in computer vision applications to correct distortions and obtain accurate measurements from images. In Electrohub, camera calibration was performed using OpenCV's camera calibration toolbox.

1. Chessboard Pattern: A chessboard pattern with known dimensions was used as a calibration target. Multiple images of the chessboard were captured from different angles and distances by the camera mounted on the robotic arm's end effector.(fig 3.5)

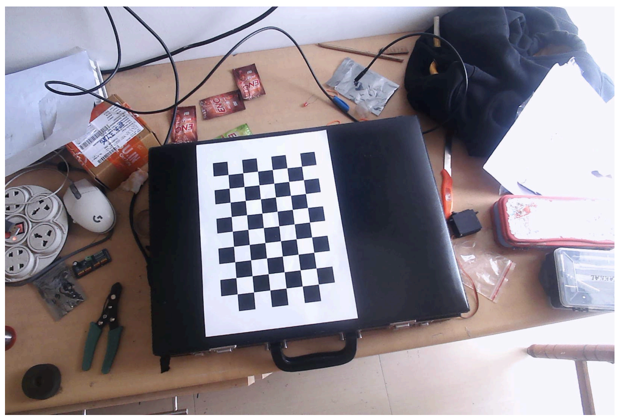
2. Image Processing: OpenCV's camera calibration toolbox processes these images to detect and extract corners of the chessboard pattern. The toolbox automatically identifies calibration points based on the known dimensions of the chessboard.
3. Intrinsic and Extrinsic Parameters: Through calibration, intrinsic parameters such as focal length, principal point, and distortion coefficients are determined. Extrinsic parameters, including the camera's position and orientation relative to the chessboard, are also calculated.

```
Calibration Matrix:
[[1.95973180e+03 0.00000000e+00 1.40002453e+03]
 [0.00000000e+00 1.95599159e+03 1.01265886e+03]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
Distortion Coefficients:
[[ 3.69746762e-02 -4.58080525e-01  1.29686535e-03 -3.18514425e-03
   1.35392375e+00]]
```

4. Calibration Result: The calibration process generates a calibration matrix and distortion coefficients, which are used to rectify images and correct distortions during subsequent image processing tasks.



**(a)** **(b)**



**(c)**

**Figure 4.9(a,b,c): Different Isometric images of Known object(chessboard 10x7) for camera calibration**

### 4.3.4 ArUco Marker

Aruco Marker Overview: Aruco markers are square markers with unique patterns that are used for marker-based augmented reality and computer vision applications. Each marker contains an encoded pattern that can be detected and recognized by algorithms, making them suitable for pose estimation and object tracking.

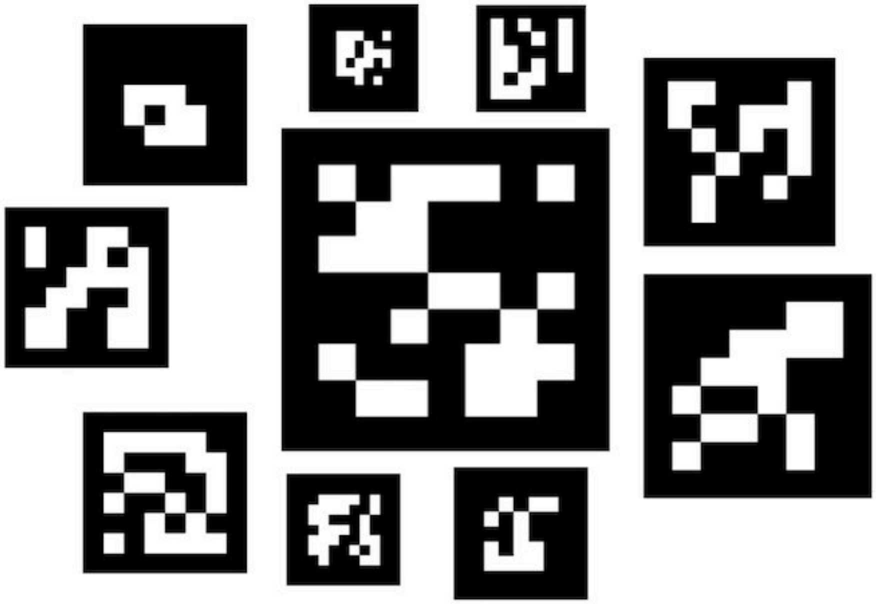


**Figure 4.10: Different types and shapes of arUco markers**

Usage in Electrohub: In Electrohub, ArUco markers are employed for pose and orientation estimation of the charging port and end effector. Since 3D stereo cameras were unavailable, ArUco markers provided a reliable alternative for marker-based localization.

**Pose and Orientation Estimation:**

1. Detection: OpenCV's ArUco module is used to detect ArUco markers in images captured by the webcam mounted on the end effector. The markers' IDs and positions in the image are determined.
2. Calibration: The camera calibration parameters obtained earlier are used to rectify and undistort the image, improving marker detection accuracy.
3. Pose Estimation: Using the detected ArUco markers' positions and known marker dimensions, pose estimation algorithms calculate the marker's 3D pose (position and orientation) relative to the camera.

4. Transformation: The pose information is transformed to the robotic arm's coordinate system, allowing precise localization of the charging port for automated plugin operations.

### 4.3.5 Automation

Automation Overview: Automation refers to the process of automatically performing tasks or operations without human intervention. In Electrohub, automation is achieved through a combination of hardware components and software algorithms.

**Transformation and Coordination:**

1. Transformation: Pose information obtained from camera calibration and ArUco marker detection is transformed into the robotic arm's coordinate system. This transformation ensures accurate positioning and orientation of the end effector and charging port relative to the arm.
2. Coordination: The transformation data is utilized to coordinate the movements of multiple components, including servo motors controlled by the PCA9685 driver and motors of the four-wheel cart controlled by the L298N motor driver. This coordination enables precise maneuvering of the robotic arm for automated charging operations.

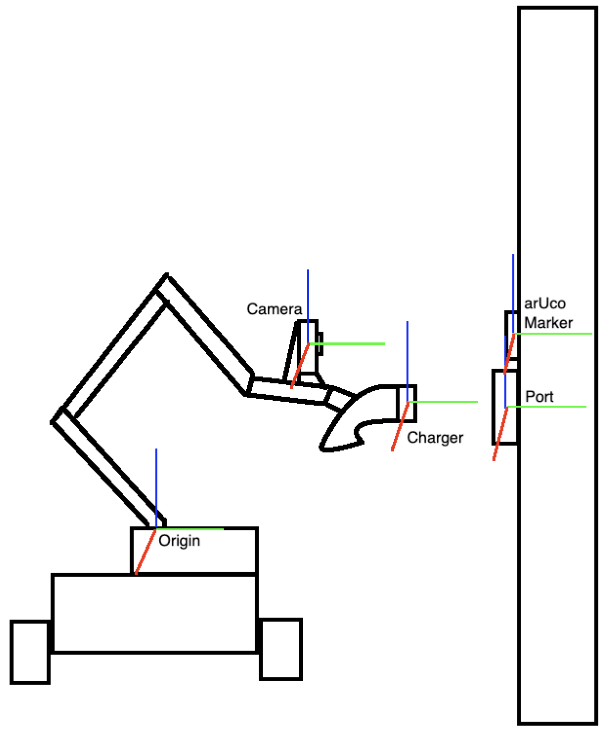



**Figure 4.11: Simple visualization of the complete prototype system**

**Transformation and Coordination:**

1. Transformation Calculations: The automation process in Electrohub involves calculating multiple transformations to ensure precise positioning and orientation of the end effector and charging port.(*general example figure 4.3.7)
   - Toe (Transformation from Origin to End Effector): This transformation defines the position and orientation of the end effector relative to the base of the robotic arm, which serves as the origin in the world frame.
   - Tec (Transformation from End Effector to Camera): Manual calculation determines the transformation from the end effector to the camera mounted on it, providing the camera's position and orientation.
   - Tca (Transformation from Camera to ArUco Marker): OpenCV functions are utilized to calculate this transformation, determining the position and orientation of the ArUco marker relative to the camera.

- Tap (Transformation from ArUco Marker to Port): Manual calculation establishes the transformation from the ArUco marker to the charging port, defining the port's position and orientation relative to the marker.

2. Overall Transformation (Top):
    - The combination of Toe, Tec, Tca, and Tap transformations yields the overall transformation (Top) from the origin to the port location. This comprehensive transformation encompasses the entire chain of transformations needed to accurately position the robotic arm for charging operations.

$$\mathrm{T}oe \times \mathrm{T}ec \times \mathrm{T}ca \times \mathrm{T}ap = \mathrm{T}op$$

3. Servo Motor Positioning:
    - By leveraging the calculated transformation (Top), Electrohub determines the precise position and orientation required for the end effector to plugin the charger into the port autonomously.
    - Servo motors' positions are calculated based on Top, ensuring the robotic arm's end effector aligns perfectly with the charging port for seamless plugin operations.

By integrating transformation calculations, servo motor control, and automation techniques, Electrohub achieves highly accurate and efficient charging operations without human intervention. This systematic approach ensures optimal performance and reliability in charging electric vehicles.

# CHAPTER 5

# CONCLUSION AND FUTURE SCOPE

## 5.1 CONCLUSION

In conclusion, the Electrohub project represents a significant advancement in electric vehicle charging infrastructure, offering a wired autonomous docking system that streamlines the charging process while reducing human intervention. By integrating robotics, computer vision, and machine learning technologies, Electrohub demonstrates the potential to revolutionize the way electric vehicles are charged in real-world scenarios.

Despite its promising potential, the Electrohub project encountered several challenges throughout its development. One of the primary difficulties arose from the machine learning model's struggle to accurately detect vehicles and locate charging ports, leading to inconsistencies in charging port alignment. Additionally, resource constraints posed limitations on the availability of hardware components and computing resources, hindering the project's progress. The complexity of Robot Operating System (ROS) programming presented another obstacle, requiring significant time and effort to navigate effectively. Furthermore, selecting an appropriate design for the robotic arm and charger end effector proved challenging, as it necessitated balancing functionality, stability, and manufacturability.

Originally envisioned as a stationary charging station, the project's requirements evolved necessitating a change in approach. To address this, the base of the robotic arm was modified to a four-wheeled platform, enabling mobility and adaptability to various charging scenarios. This strategic shift allowed Electrohub to overcome spatial constraints and cater to dynamic charging needs in real-world environments. Despite these challenges, the team's perseverance and collaborative efforts ultimately led to the successful development of Electrohub—an innovative solution poised to revolutionize electric vehicle charging infrastructure.

In real-life scenarios, Electrohub could be implemented in parking lots, residential complexes, commercial areas, and public charging stations, catering to the growing demand for electric vehicle charging solutions. By leveraging existing resources and infrastructure, organizations and individuals can easily deploy Electrohub to enhance EV charging accessibility and efficiency, ultimately contributing to the widespread adoption of sustainable transportation practices.

In essence, Electrohub exemplifies a promising step forward in the electrification of transportation, offering a practical and innovative solution to address the evolving needs of electric vehicle owners and charging infrastructure providers alike. Through continued refinement and deployment, Electrohub has the potential to play a pivotal role in shaping the future of sustainable mobility.

## 5.2 FUTURE SCOPE

The presented system is a vision-guided and robot-based automatic charging station for electric and hybrid vehicles. It aims to automate the conductive fast charging process, ensuring efficient use of charging spaces.

The system combines shape-based template matching and stereo cameras to identify the charging port type and estimate its position and orientation. The robot then approaches and plugs in the charger cable, using force sensors to detect and adjust for any misalignment.

Future improvements include enhancing connector plug detection, automating marker-less calibration, and integrating a linear actuator to open and close the charging port lid. The system will be tested on a real electric vehicle in both garage and outdoor environments, with communication between the vehicle and charging station under development for integration with autonomous parking functions.

This automated charging solution addresses the issue of occupied charging stations when not in use, contributing to the widespread adoption of electric mobility by providing a convenient and efficient charging experience.

**References**

[1] 3D Vision Guided Robotic Charging Station for Electric and Plug-in Hybrid Vehicles Justinas Mišeikis, M. Rüther, +2 authors Helmut Brunner Published in **arXiv.org** 1 March 2017 Engineering, Environmental Science TLDR

[2] Saputra, Hendri & Mohd Nor, Nur Safwati & Rijanto, Estiko & Md zain, M. Z. & Darus, Intan & Yazid, Edwar. (2023). A review of robotic charging for electric vehicles. International Journal of Intelligent Robotics and Applications. 8. 10.1007/s41315-023-00306-x.

[3] Noh, Donghee & Kim, Seonhyeong & Kim, Seojeong & Kim, Donghoon & Choi, Kyoungho & Choi, Juhwan & Park, Keunho. (2023). Docking Method for Electric Vehicle Charging Terminal Using Monocular Camera. 10.21203/rs.3.rs-3180077/v1.

[4] Pengkun, Quan & Lou, Ya'nan & Lin, Haoyu & Liang, Zhuo & Di, Shichun. (2021). Research on Fast Identification and Location of Contour Features of Electric Vehicle Charging Port in Complex Scenes. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3092210.

[5] Mahaadevan, Vc & Rajamanickam, Narayanamoorthi & Dutta, Sankhaddep & Dooly, Gerard. (2024). AViTRoN: Advanced Vision Track Routing and Navigation for Autonomous Charging of Electric Vehicles. IEEE Access. PP. 1-1. 10.1109/ACCESS.2024.3355018.

[6] Zhou, Y., He, Y., Yan, Y., Li, F., Li, N., Zhang, C., Lu, Z., & Yang, Z. (2023). Autonomous charging docking control method for unmanned vehicles based on vision and infrared. Journal of Physics: Conference Series, 2584, 012065. doi:10.1088/1742-6596/2584/1/012065

[7] Ulaszewski, Maciej & Janowski, Robert & Janowski, Andrzej. (2022). Application of computer vision to egg detection on a production line in real time.. ELCVIA Electronic Letters on Computer Vision and Image Analysis. 20. 113-143. 10.5565/rev/elcvia.1390.

# Appendix

### Pose Estimation

```
from imutils.video import VideoStream

import argparse

import imutils

import time

import cv2

import sys

import json

import numpy as np

ap = argparse.ArgumentParser()

ap.add_argument("-t", "--type", type=str,

            default="DICT_ARUCO_ORIGINAL",

            help="type of ArUCo tag to detect")

args = vars(ap.parse_args())



ARUCO_DICT = {

  "DICT_4X4_50": cv2.aruco.DICT_4X4_50,
```

```
"DICT_4X4_100": cv2.aruco.DICT_4X4_100,

"DICT_4X4_250": cv2.aruco.DICT_4X4_250,

"DICT_4X4_1000": cv2.aruco.DICT_4X4_1000,

"DICT_5X5_50": cv2.aruco.DICT_5X5_50,

"DICT_5X5_100": cv2.aruco.DICT_5X5_100,

"DICT_5X5_250": cv2.aruco.DICT_5X5_250,

"DICT_5X5_1000": cv2.aruco.DICT_5X5_1000,

"DICT_6X6_50": cv2.aruco.DICT_6X6_50,

"DICT_6X6_100": cv2.aruco.DICT_6X6_100,

"DICT_6X6_250": cv2.aruco.DICT_6X6_250,

"DICT_6X6_1000": cv2.aruco.DICT_6X6_1000,

"DICT_7X7_50": cv2.aruco.DICT_7X7_50,

"DICT_7X7_100": cv2.aruco.DICT_7X7_100,

"DICT_7X7_250": cv2.aruco.DICT_7X7_250,

"DICT_7X7_1000": cv2.aruco.DICT_7X7_1000,

"DICT_ARUCO_ORIGINAL": cv2.aruco.DICT_ARUCO_ORIGINAL,

"DICT_APRILTAG_16h5": cv2.aruco.DICT_APRILTAG_16h5,

"DICT_APRILTAG_25h9": cv2.aruco.DICT_APRILTAG_25h9,

"DICT_APRILTAG_36h10": cv2.aruco.DICT_APRILTAG_36h10,
```

```
  "DICT_APRILTAG_36h11": cv2.aruco.DICT_APRILTAG_36h11
}


detectgray = True
drawaxes = True


if ARUCO_DICT.get(args["type"], None) is None:
  print("[INFO] ArUCo tag of '{}' is not supported".format(
      args["type"]))
  sys.exit(0)


print("[INFO] detecting '{}' tags...".format(args["type"]))


dictionary = cv2.aruco.getPredefinedDictionary(cv2.aruco.DICT_4X4_250)
arucoParams =  cv2.aruco.DetectorParameters()


print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()
time.sleep(2.0)
```

```
with np.load('new_calibration_data.npz') as X:
  mtx, dist, _, _ = [X[i] for i in ('mtx','dist','rvecs','tvecs')]


frame = vs.read()
h, w = frame.shape[:2]


newcameramtx, roi = cv2.getOptimalNewCameraMatrix(mtx, dist, (h, w), 0, (h, w))
mapx, mapy = cv2.initUndistortRectifyMap(mtx, dist, None, newcameramtx, (w, h), cv2.CV_32FC1)
x, y, w1, h1 = roi
yh1 = y + h1
xw1 = x + w1


while True:
  frame = vs.read()


  dst1 = cv2.remap(frame, mapx, mapy, cv2.INTER_LINEAR)
  frame = dst1[y:yh1, x:xw1]
```

```
  if detectgray:
      gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
      (corners, ids, rejected) = cv2.aruco.detectMarkers(gray, dictionary, parameters=arucoParams)
  else:
      (corners, ids, rejected) = cv2.aruco.detectMarkers(frame, dictionary, parameters=arucoParams)

  if len(corners) > 0:
      if drawaxes:
          for i in range(0, len(ids)):
              rvec, tvec, markerPoints = cv2.aruco.estimatePoseSingleMarkers(corners[i], 0.02, mtx, dist)
              cv2.drawFrameAxes(frame, mtx, dist, rvec, tvec, 0.02)

              # Print the pose estimation and orientation
              print(f"Marker ID: {ids[i]}")
              print("Rotation Vector:")
              print(rvec)
```

```
            print("Translation Vector:")
            print(tvec)

        ids = ids.flatten()
        cv2.aruco.drawDetectedMarkers(frame, corners, ids)

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF

    if key == ord("q"):
        break

cv2.destroyAllWindows()
vs.stop()
```

Inverse Kinematic:

```
import ikpy

import numpy as np

import ikpy.utils.plot as plot_utils

from ikpy.chain import Chain

from ikpy.link import OriginLink, URDFLink

import matplotlib.pyplot as plt

from mpl_toolkits.mplot3d import Axes3D

import time



my_chain = ikpy.chain.Chain.from_urdf_file("lol.urdf")



#target_position = [0.4, 0.4, 0.4]

#print("The angles of each joints are : ", my_chain.inverse_kinematics(target_position))



target_position_1 = [0.4, 0.4, 0.4]

print("The angles of each joint for target 1 are:", my_chain.inverse_kinematics(target_position_1))
```

```python
target_position_2 = [0,0,2]

print("The angles of each joint for target 2 are:", my_chain.inverse_kinematics(target_position_2))



fig = plt.figure()

ax = fig.add_subplot(111, projection='3d')



plot_1 = my_chain.plot(my_chain.inverse_kinematics(target_position_1), ax)



# Plotting target_position_2

plot_2 = my_chain.plot(my_chain.inverse_kinematics(target_position_2), ax)



for line in ax.lines:

   if line.get_label() == 'my_chain':

       line.set_visible(False)

plt.xlim(-1.5, 1.5)

plt.ylim(-1.5, 1.5)

#plt.zlim(-3, 3)

plt.show()
```

```
#ax = matplotlib.pyplot.figure().add_subplot(111, projection='3d')



#my_chain.plot(my_chain.inverse_kinematics(target_position), ax)

#matplotlib.pyplot.show()
```

mhp40

ORIGINALITY REPORT

| 3% | 3% | 0% | 3% |
| --- | --- | --- | --- |
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
| --- | --- | --- |
| 1 | stackoverflow.com<br>Internet Source | 2% |
| 2 | Submitted to IIT Delhi<br>Student Paper | <1% |
| 3 | Submitted to SRM University<br>Student Paper | <1% |
| 4 | python.hotexamples.com<br>Internet Source | <1% |
| 5 | petamind.com<br>Internet Source | <1% |
| 6 | www.coursehero.com<br>Internet Source | <1% |
| 7 | en.wikibooks.org<br>Internet Source | <1% |
| 8 | Submitted to Temple University<br>Student Paper | <1% |
| 9 | 123dok.org<br>Internet Source | <1% |