TRASHOBOT

DSN4096-CAPSTONE PROJECT PHASE-II

Phase – II Report

Submitted by

SOUBHIK PAILAN
(20BAI10276)

in partial fulfillment for the award of the degree
of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
VIT BHOPAL UNIVERSITY

SEHORE, MADHYA PRADESH - 466114

MAY 2024

VIT BHOPAL UNIVERSITY, KOTHRIKALAN, SEHORE MADHYA PRADESH – 466114

BONAFIDE CERTIFICATE

Certified that this project report titled **"TRASHOBOT - A TRASH PICKING ROBOT"** is the bonafide work of "**SOUBHIK PAILAN(20BAI10276)"** who carried out the project work (DSN4096- Capstone Project phase-II) under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR
Dr Pradeep Kumar Mishra,
Senior Assistant Professor (Gr-2),
School of Computing Science and Engineering,
VIT BHOPAL UNIVERSITY.

PROJECT GUIDE
Dr Pranshu Pranjal
Assistant Professor (Gr-1) ,
School of Computing Science and Engineering,
VIT BHOPAL UNIVERSITY.

The DSN4096 - Capstone Project phase-II Viva Examination was held on 11.03.2024

ACKNOWLEDGEMENT

I extend my deepest gratitude to all those who have contributed to the success of this project.

I would like to thank my internal guide Dr. Pranshu Pranjal, for continually guiding and actively participating in our project, and giving valuable suggestions to complete the project works.

I wish to express our heartfelt gratitude to Dr. Pradeep Kumar Mishra , Program Chair - Division of AI and Ml, School of Computing Science and Engineering for his valuable support and encouragement in carrying out this work.

I wish to express our heartfelt gratitude to Dr. S. Poonkuntran, Professor and Dean, School of Computing Science and Engineering for his valuable support and encouragement in carrying out this work.

I would like to thank all the technical and teaching staff of the School of Computing Science and Engineering, who extended directly or indirectly all support.

My appreciation extends to the Raspberry Pi Foundation, whose computing platforms played a fundamental role in my research and execution phases.

Last, but not least, I am deeply indebted to my parents who have been the greatest support for successful completion of this project. Their encouragement and belief in my abilities were vital to my perseverance.

(Soubhik Pailan)

LIST OF FIGURES

ABSTRACT

The primary goal of the project is to design a trash-picking robot. The “Trashobot” is a breakthrough in household cleaning technology, integrating a 6 DOF robotic arm and advanced computer vision to revolutionize trash removal. We have developed a robotic model that can be operated either manually or automated to do a certain task. The instructions given to the robotic jaw for collecting waste and dry materials is done by identifying using installed sensor and the machine learning algorithm. In the robot the electronic system is centered around the Raspberry Pi 4B, which acts as the brain of Trashobot, processing data from the Raspberry Pi Camera Module 3 for object detection. The SG90 Micro servo Motor aids in fine-tuning camera movements, enhancing the accuracy of object identification. To power its operations, the robot is equipped with batteries, eliminating the need for additional fuels. 6 DOF arm uses 6 servo motor and it can be controlled both manually as well as automatically by the Raspberry Pi. After finalizing the design, 3D printed parts for the robot car base and the arm were printed, assembled and tested for mechanical integrity and functionality. The Trashbot achieved an average accuracy of ~88.89% (identifying the trash accuracy), and the average confidence while correct prediction of trash using sensors is ~72.14%.

TABLE OF CONTENTS

CHAPTER 1

PROJECT DESCRIPTION AND OUTLINE

1.1 INTRODUCTION

Waste management is a critical challenge worldwide, with a growing emphasis on developing innovative solutions to handle waste more effectively and sustainably. In India, the improper handling of waste poses significant risks to public health, sanitation, and wildlife. To address this issue, the Trashobot project presents a novel approach to household waste management using advanced robotics and computer vision technology.

Trashobot is a state-of-the-art robotic system designed to autonomously identify, collect, and dispose of household trash. Unlike conventional household cleaning robots that primarily vacuum dust and small debris, Trashobot has a precise and targeted approach – one that could identify and selectively pick up trash, instead of aimlessly sweeping around like existing household cleaning robots. Trashobot integrates advanced computer vision, machine learning and a precise robotic picking motion to achieve its goal.

1.2 MOTIVATION FOR WORK

The Trashobot project is motivated by the increasing demand for efficient household waste management solutions. In today's world, domestic waste is a major contributor to environmental pollution, and existing cleaning robots often lack the precision needed to address this issue effectively. Trashobot seeks to fill this gap by offering a more targeted approach to trash removal, addressing the limitations of current technologies.

I envisioned a more advanced robot capable of precisely targeting and selectively picking up trash. This approach aims to address the inefficiencies of traditional robots, which often sweep aimlessly. Trashobot's precise and targeted methodology seeks to enhance household cleaning by leveraging advanced computer vision and machine learning technologies, providing a more effective solution for waste management in domestic settings.

1.3 PROBLEM STATEMENT

The project addresses inefficiencies in current waste management practices and the lack of autonomous solutions for environmental cleanup. Traditional household cleaning robots lack the precision and specificity needed for effective waste collection, often missing larger debris or operating inefficiently. Trashobot aims to solve this problem by offering an autonomous, targeted solution for household waste, leveraging advanced computer vision and machine learning to improve the accuracy and efficiency of waste collection.

1.4 OBJECTIVE OF THE WORK

The primary objective of the Trashobot project is to develop an innovative robotic solution for household waste management. Creating a trash-picking robot equipped with advanced features, such as a 6DOF robotic arm and computer vision. The specific goals include:

- Improving Efficiency: Enhancing the accuracy and efficiency of household cleaning by targeting and selectively removing trash.
- Automation: Developing an autonomous system that can function independently or through remote commands.
- Sustainability: Contributing to environmental sustainability by addressing waste management issues in domestic settings.

1.5 SUMMARY

In summary, trashobot is introduced as a pioneering household cleaning robot designed to address waste management inefficiencies. The motivation behind the project stems from a desire to improve upon existing cleaning robots, such as the Roomba, by creating a more targeted and efficient solution. The project addresses the lack of autonomous solutions for environmental cleanup, with the objective of developing an efficient, autonomous robot that can identify and collect trash, thereby promoting cleaner environments. The problem statement and objectives highlight the need for innovation in waste management, particularly in domestic settings.

CHAPTER 2

RELATED WORK INVESTIGATION

2.1 EXISTING APPROACHES/METHODS

The evolution of household cleaning robots has largely been represented by robotic vacuums, such as Roomba. These robots have automated floor cleaning, primarily by vacuuming dust and small debris.

The Roomba, developed by iRobot, is an autonomous robotic vacuum cleaner known for its ability to navigate rooms and clean floors. It uses a combination of sensors and algorithms to map out its surroundings, allowing it to avoid obstacles and efficiently clean various floor surfaces. They operate based on a predefined cleaning pattern or can adapt its path based on detected dirt levels, making it a versatile cleaning tool.

While existing household cleaning robots have focused primarily on vacuuming, the advancements in computer vision and robotics offer a unique opportunity to address larger waste items. Trashobot's innovative design, which merges computer vision and robotic manipulation, addresses this gap, positioning it as a pioneering solution in household cleaning technology.

2.2 PROS AND CONS OF THE STATED APPROACHES/ METHODS

Pros of existing solutions:

- **Market Presence:** Existing cleaning robots, such as Roomba are a recognized brand, which ensures customer trust and ease of accessibility. Its widespread adoption means users can easily find support and accessories.

- **Simple Design:** Many current household cleaning robots feature straightforward design makes it user-friendly. It requires minimal setup and can be easily operated by individuals of all technical backgrounds.

- **Automation:** Most of these robots operates autonomously, freeing up time for users. It can be scheduled to clean at specific times, making it convenient for daily use.

Cons of existing solutions:

- **Limited Targeting**: Many existing cleaning robot lack the precision to handle specific trash types effectively. It's primarily designed for vacuuming and may struggle with larger or irregularly shaped debris.

- **Random Patterns:** Some cleaning robots follow a less efficient, random cleaning pattern. This can result in missed spots or redundant cleaning, which may waste time and energy.

- **Maintenance:** Regular maintenance is necessary to keep them functioning optimally. This includes cleaning filters and brushes, which can be time-consuming and inconvenient for some users.

- **Navigation Issues:** They sometimes struggles with obstacles, such as getting stuck under furniture or failing to clear certain barriers. This can interrupt the cleaning process and require user intervention.

CHAPTER-3

REQUIREMENT ARTIFACTS

3.1 INTRODUCTION

In the development of Trashobot, it was essential to define and document the requirements that would guide its design and functionality. Let us look at the requirement artifacts that were created during the early stages of the project, providing a comprehensive overview of the system's functional and non-functional needs. These artifacts serve as the foundation for the subsequent design and implementation phases, ensuring that Trashobot meets the expected standards and user expectations.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

<u>The primarily hardware requirements include:</u>

- **Single Shaft L-Shaped 60 RPM Battery Operated Motor (4x)**

  These motors provide the primary propulsion for the robot. The L-shaped design allows for easy mounting, and the 60 RPM (Rotations Per Minute) speed ensures adequate mobility. They are battery-operated, making them suitable for mobile applications.

- **Wheels Diameter -65mm, width -15 mm**

  The wheels determine the mobility and maneuverability of the robot. With a diameter of 65mm and a width of 15mm, they strike a balance between stability and agility, enabling the robot to navigate various terrains effectively.

- **MG996R Servo Motor (x3)**

  These servo motors are used to control the movement of a 6 Degrees of Freedom (DOF) robotic arm. With a torque output of 9.4 kg-cm, they provide sufficient power to lift and manipulate objects.

- **Raspberry Pi 4B 4GB RAM**

  The Raspberry Pi serves as the brain of the robot, handling computation, decision-making, and communication tasks. With its 4GB RAM, the Raspberry Pi 4B is capable of running complex algorithms for navigation, object detection, and control.

- **SG90 Micro Servo Motor (x4)**

  These micro servo motors are employed for precise movements, such as controlling the joints of the robotic arm and adjusting the orientation of the camera. While they produce lower torque (1.2 kg-cm), their compact size and precision make them ideal for fine adjustments.

- **4-Channel DC Motor Driver Board**

  This motor driver board controls the four motors responsible for the robot's locomotion. It regulates the voltage and current supplied to the motors, allowing for smooth and efficient operation.

- **16-Channel 12-bit PWM Servo Driver I2C Interface**

  The PWM (Pulse Width Modulation) servo driver enables precise control of the seven servo motors used in the robot. With 16 channels and 12-bit resolution, it offers flexibility and accuracy in positioning the robotic arm and camera.

- **Raspberry Pi Camera Module 3**

  The camera module provides visual perception for the robot, allowing it to detect obstacles, identify objects, and navigate its environment. It interfaces seamlessly with the Raspberry Pi, enabling real-time image processing and analysis.

- **DC-DC Buck Converter**

  The buck converter regulates the voltage supplied to various components, ensuring stable and reliable operation. It converts the input voltage from the power source (e.g., the power bank) to the appropriate levels required by different components.

- **12-watt Power Bank**

  The power bank serves as the primary power source for the robot, supplying electricity to all components. With its 12-watt capacity, it provides sufficient power for extended operation periods, making the robot portable and self-sufficient.

The primary Software requirements include:

- **Autodesk Fusion 360 for 3D modelling**

  Autodesk Fusion 360 is utilized for 3D modeling, enabling precise and detailed designs for Trashobot's mechanical components.

- **TensorFlow lite for designing ML algorithms**

  TensorFlow Lite is employed to develop machine learning algorithms, enhancing Trashobot's ability to detect and classify various types of trash.

- **Debian based Linux for Raspberry Pi**

  Raspbian Bullseye, a Debian-based Linux distribution, is installed on the Raspberry Pi, providing a stable and reliable operating system for controlling Trashobot's functions.

- **Integrated Developers Environment for Raspberry Pi**

  Thonny, an Integrated Development Environment (IDE), is used for programming and debugging tasks on the Raspberry Pi, ensuring efficient code development and optimization for Trashobot's operations.

3.3 SUMMARY

We looked at the comprehensive overview of the foundational requirements and discussed about the primary hardware and software requirements that shaped the trajectory of the Trashobot project. From the selection of hardware components to the definition of software tools and technologies, these provide a comprehensive overview of the system's functional and non-functional needs move forward with confidence to design and implement the various components that will bring Trashobot to life.

CHAPTER-4

DESIGN METHODOLOGY AND ITS NOVELTY

4.1 METHODOLOGY AND GOAL

The design methodology of Trashobot involved iterative development with a focus on achieving a balance between functionality and sustainability. The goal was to create a robot capable of effectively identifying and picking up trash autonomously.

Brainstorming and Conceptualization: This initial phase involved ideation and planning of the robot's design and functionalities, focusing on its capability to navigate and pick up trash in household environments.

Designing the Robot Car Base: The base of the robot was designed for stability and maneuverability in varied indoor terrains.

Building a Thermocol Prototype: A lightweight thermocol prototype of the robot car base and the 6DOF robotic arm was constructed to visualize and refine the design.

Designing the 6 DOF Robotic Arm: The robotic arm was designed to achieve precise and adaptable movements for picking up trash of different sizes and shapes.

4.2 FUNCTIONAL MODULES DESIGN AND ANALYSIS

The design consists of several key modules:

Robot Car Module: Four-wheeled design for stable and agile navigation in household environments, taking into account obstacles and doorways.

6 DOF Robotic Arm Module: Six degrees of freedom for precise and versatile movement capabilities. High torque servo motors ensure efficient lifting and manipulation of objects.

Computer Vision Module: Utilizes a high-resolution camera for real-time image capture and analysis. Employs machine learning algorithms for object detection and classification.

Each module was carefully designed to ensure efficient operation and integration. The Trashobot needs to fit in domestic environments and pass through doors, underneath of chairs, it's crucial that the design is compact and lightweight.

4.3 SOFTWARE ARCHITECTURAL DESIGNS

The software architecture is centered around a Raspberry Pi controller, which processes input from sensors and camera, managing the overall operation of the robot including navigation and object manipulation.

Subsystem Services

Navigation Subsystem:

Four-wheeled robot car with precise control and obstacle detection sensors for navigating household environments. Sophisticated algorithms for path planning and obstacle avoidance ensure safe and efficient traversal.

Trash Detection Subsystem:

High-resolution camera and advanced image processing algorithms for detecting and classifying waste. Machine learning techniques, such as CNNs, enable accurate identification of trash items.

Trash Picking Subsystem:

Six-degree-of-freedom robotic arm with high-torque servo motors for precise trash manipulation. Computer vision feedback and proprioceptive sensors guide the arm's movements during picking operations.

Integration and Coordination Subsystem:

Facilitates seamless communication and coordination between the navigation, trash detection, and trash picking subsystems. Utilizes a centralized control unit, such as Raspberry Pi 4, to orchestrate the actions of each subsystem based on real-time sensor data and environmental feedback.

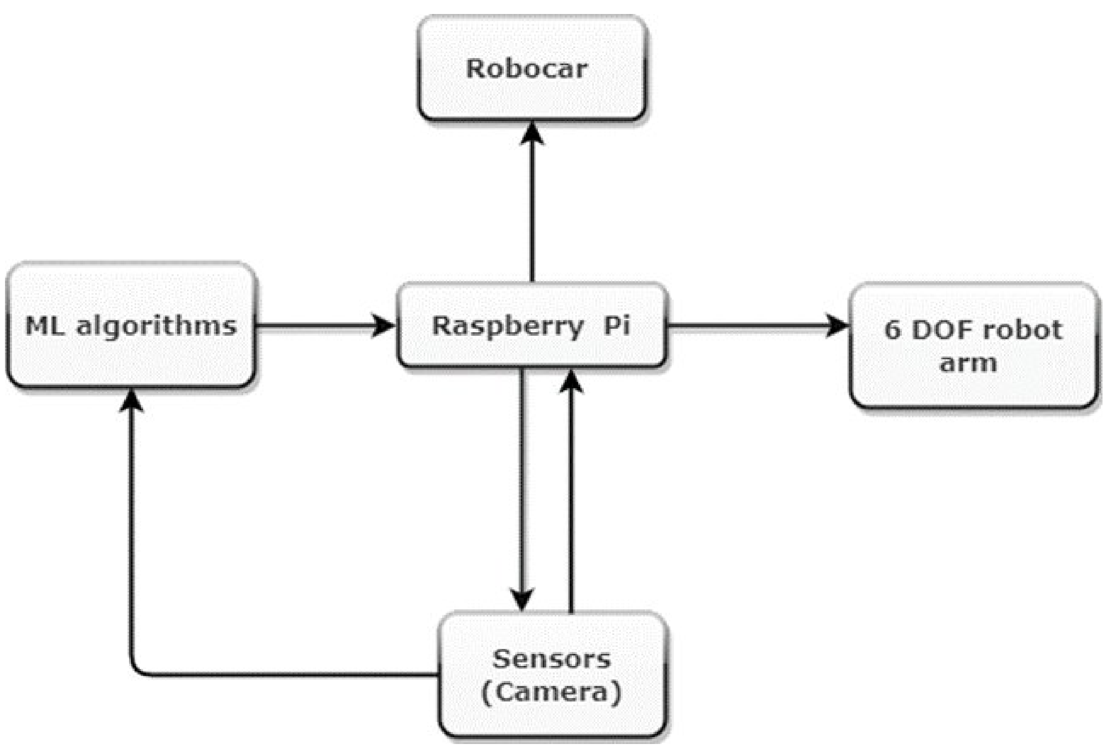


Fig 1 : System Architecture Diagram

4.4 SUMMARY

We saw the innovative design methodology of Trashobot, highlighting its unique features and functional capabilities.Through meticulous design, development, and integration of subsystems, Trashobot demonstrates its ability to navigate household environments, detect various types of waste, and pick up trash items with precision and efficiency. The project underscores the potential of robotics and artificial intelligence to address real-world challenges, such as waste management, in a sustainable and impactful manner. Moving forward, the success of Trashobot paves the way for further research and development in the field of domestic robotics, with the ultimate goal of creating cleaner, healthier, and more sustainable living spaces

CHAPTER-5

TECHNICAL IMPLEMENTATION AND ANALYSIS

5.1 OUTLINE

I. Introduction to Technical Implementation:

A. Overview of the software and hardware components used in Trashobot:

- Trashobot utilizes a combination of hardware and software components to achieve its objectives.
- Hardware components include a robot car base, a 6 DOF robotic arm, sensors, and a central control unit (e.g., Raspberry Pi).
- Software components consist of algorithms for navigation, trash detection, and trash picking, implemented using programming languages such as Python.

B. Programming languages, frameworks, and tools employed:

- Python is the primary programming language used for implementing Trashobot's software.
- Libraries and frameworks such as libcamera and TensorFlow are utilized for computer vision and machine learning tasks.

II. Hardware Implementation:

A. Components of Trashobot:

- The robot car base comprises four wheels, motors, and chassis, providing mobility and stability.
- The 6 DOF robotic arm consists of servo motors, gripper.
- Sensors including cameras and proximity sensors.

B. Assembly process:

- After finalizing the design, 3D printed parts for the robot car base and the arm were assembled and tested for mechanical integrity and functionality.

- Once individual components are tested and verified, they are integrated into the complete trash-picking robot system.
- The design of the 3D-printed parts were improved iteratively and sensors were calibrated to optimize performance.

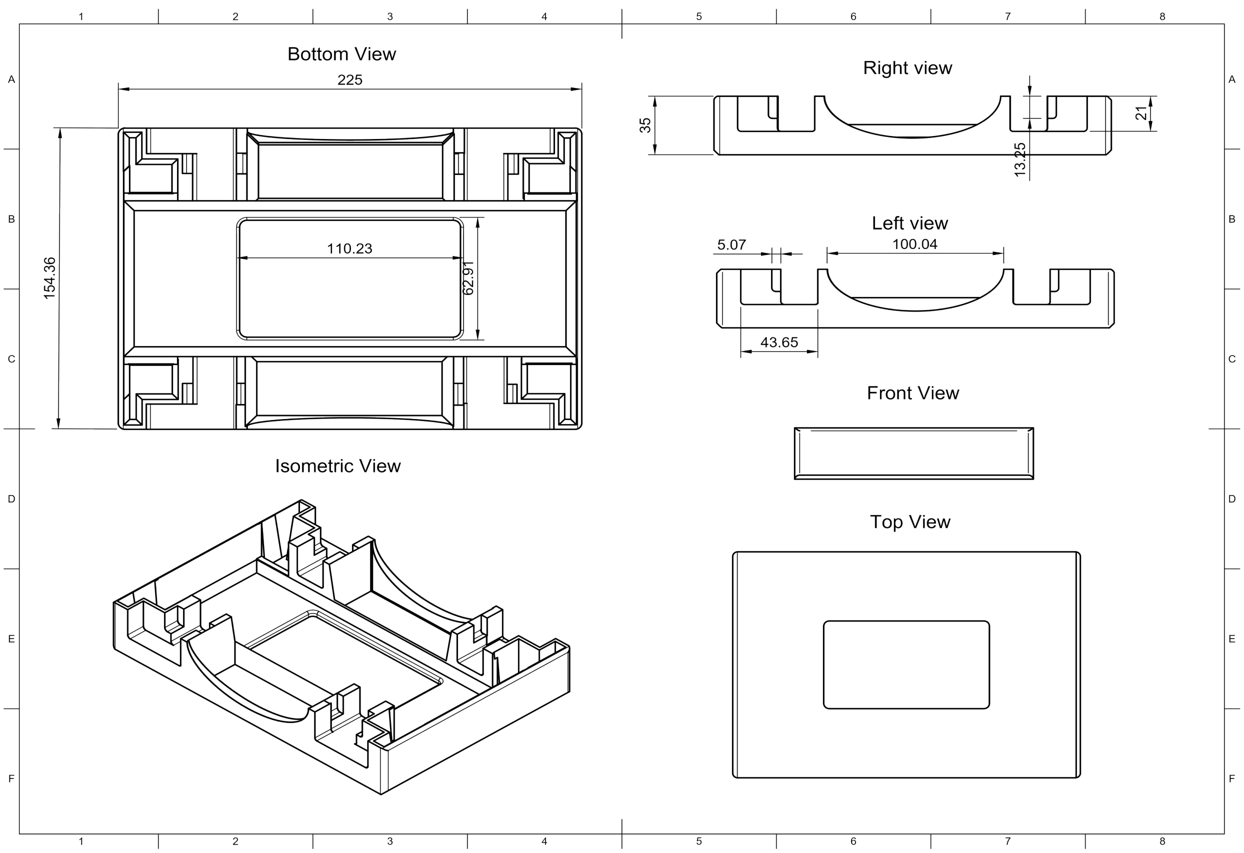


Figure 2: Design and Dimension of 3D printed robot car base.

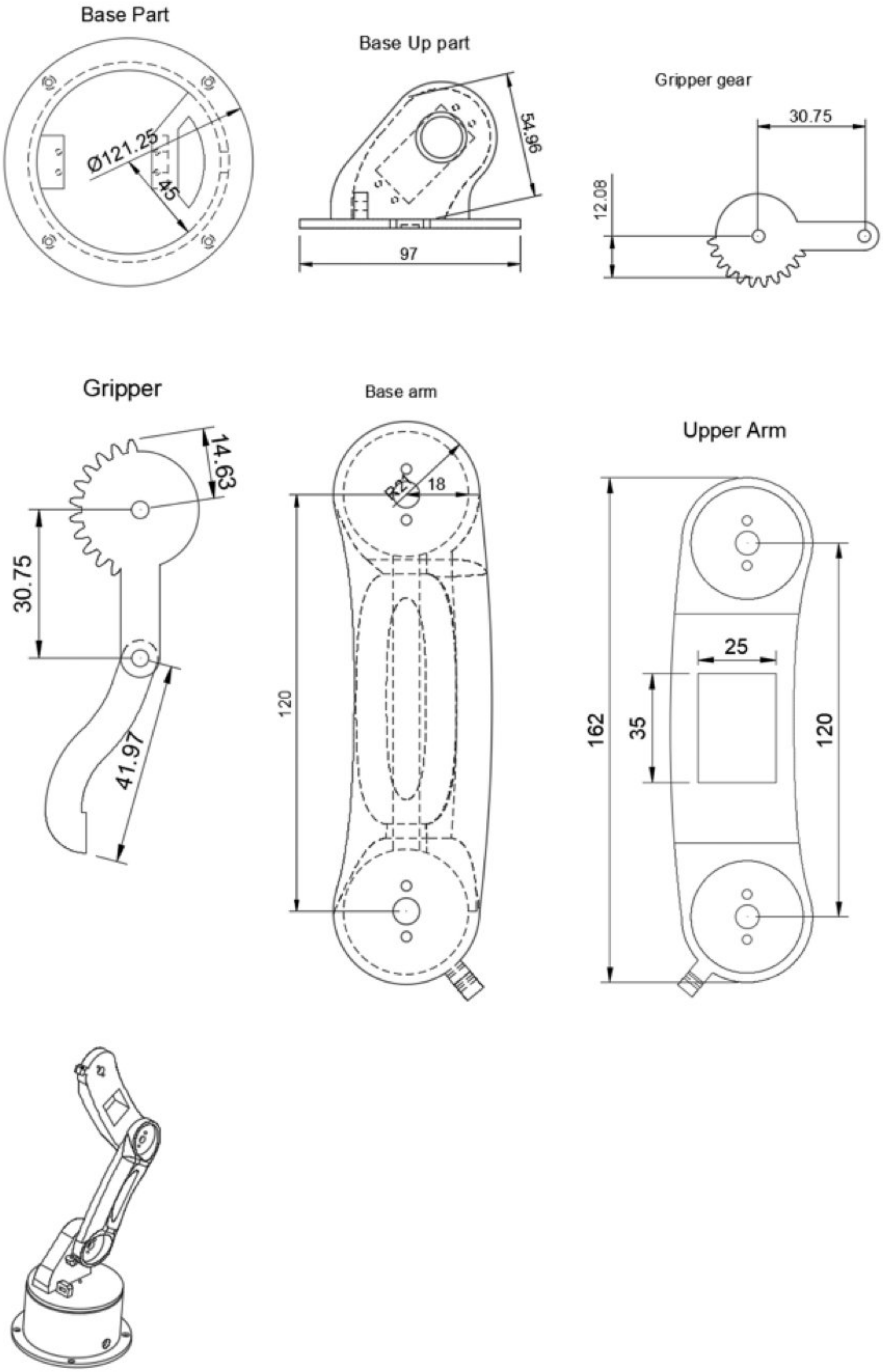


Figure 3: 6 DOF robot arm design and dimensions

III. Testing and Validation Procedures:

Overview of the testing methods employed:

- Testing procedures involve simulated real-world scenarios to assess Trashobot's capabilities under various conditions.
- Test cases are designed to evaluate navigation accuracy, trash detection reliability, and trash picking efficiency.
- Test environments include indoor settings with different layouts and obstacles.
- Scenarios simulate common household scenarios such as cluttered floors, diverse types of trash, and varying lighting conditions.

IV. Challenges and Solutions:

- **Circuit Design Errors:** Initially faced with errors in the circuit design, which affected the overall functionality of the robot.
  Solution: Conducted thorough troubleshooting and debugging of the circuit design to identify and rectify any errors.
- **Calibration of Motor Movement**: Inaccurate calibration of motor movement directions led to inefficient and incorrect maneuvers of the robot.
  Solution: Implemented precise calibration and rewiring procedures to adjust motor movement directions accurately.
- **Chassis Measurement Precision:** Inconsistent chassis measurements resulted in stability issues.
  Solution: Employed precise measurement and iterative approach to improve the components fit.

- **Servo Motor Calibration:** Servo motors failed to rotate to the correct angles, preventing the robot arm from achieving the desired movements.
  Solution: Implemented meticulous calibration procedures as well as programming precise movement sequences to adjust servo motor angles accurately.

5.2 TECHNICAL CODING AND CODE SOLUTIONS

1. **Setting up Raspberry Pi 4:** First we selected an appropriate operating system (OS) for the Raspberry Pi, such as Raspbian based on Linux Debian. It was installed onto the Raspberry Pi using a microSD card and the Raspberry Pi Imager tool. After installing the OS, the Raspberry Pi was configured to connect to the internet and enabled SSH (Secure Shell) for remote command-line access.

2. **Setting up the Development Environment:** Popular IDE like Thonny were considered based on its features , requirements of the project and ease of use.

3. **Coding the Trashobot in modules (robot car section):** Necessary packages were imported and set up the GPIO (General Purpose Input/Output) pins for interfacing with the robot's motors and sensors. The robot car was programmed to work with wireless manual control from a computer keyboard.

4. **Coding the 6 DOF Robotic Arm:** The coding process involved importing libraries for servo motor control and setting up the GPIO pins corresponding to each servo motor. Specific movement sequences were programmed to control the six degrees of freedom (DOF) of the robotic arm, including rotation, lifting, and gripping functionality. Servo motors were calibrated to ensure precise and accurate arm movements. Extensive testing and debugging were conducted to verify the functionality and reliability of the robotic arm code.

5. **Designing and 3D Printing Robot Car Body and Robotic Arm Components:**
   Detailed CAD (Computer-Aided Design) models were created for both the robot car body and the individual components of the robotic arm using Autodesk Fusion 360. Utilizing 3D printing technology, the designed models were converted into physical prototypes. Iterative testing and refinement were carried out to ensure that the 3D printed components met the desired specifications.

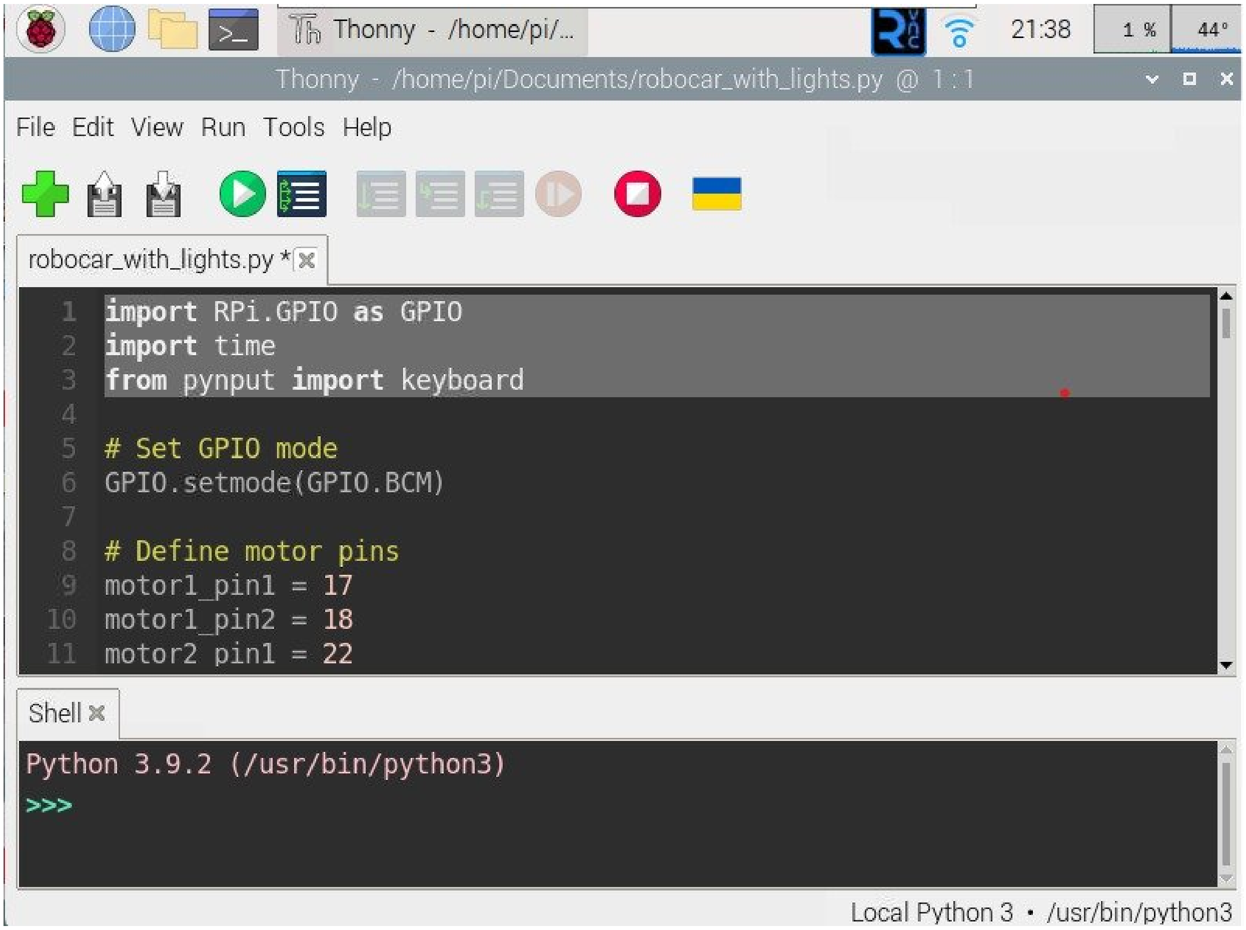


Figure 4: Coding the robot car movement functionality

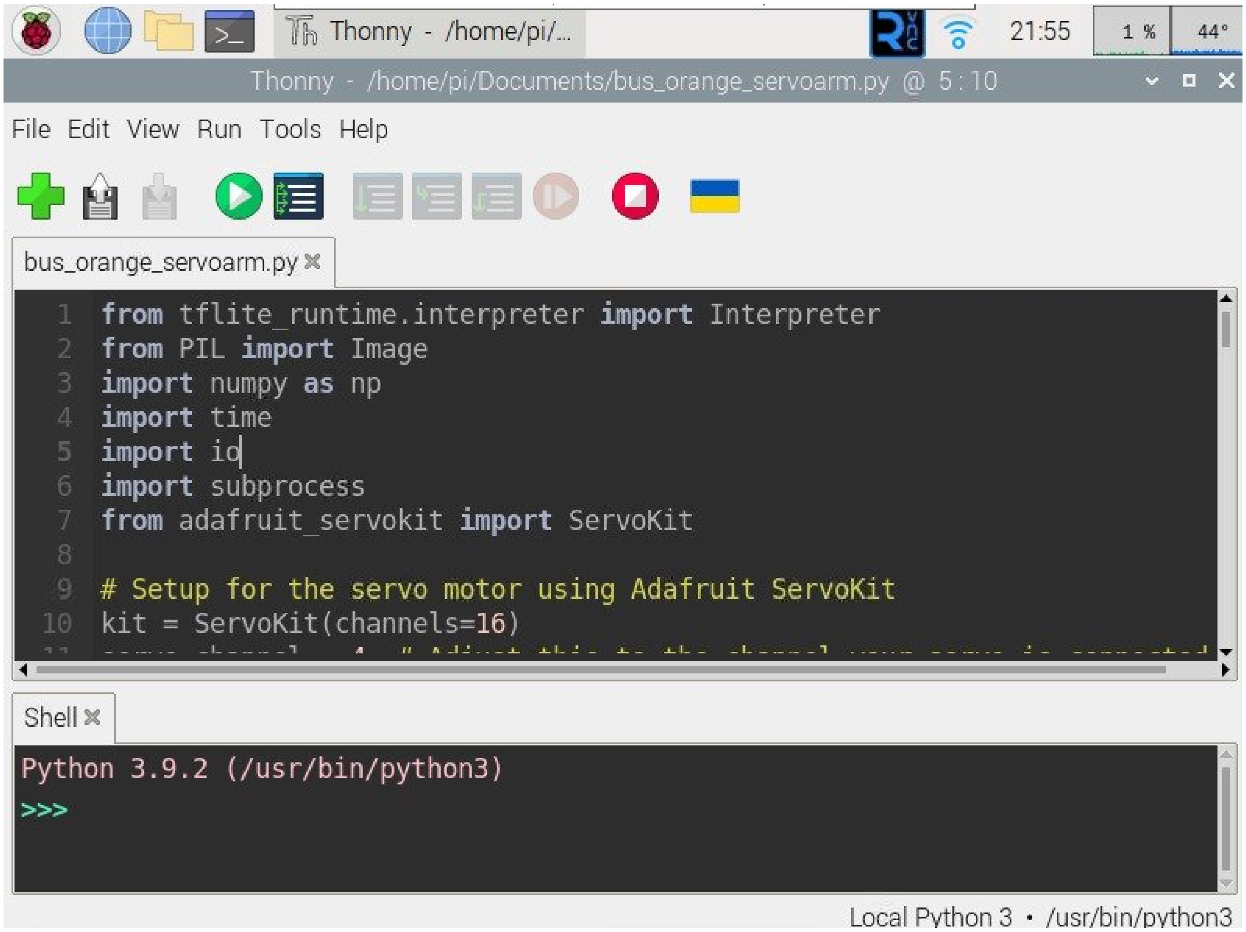


Figure 5: Programming the 6 Degree of Freedom robot arm

5.3 PROTOTYPE SUBMISSION

The development of the Trashobot prototype followed an iterative process, with each version aimed at enhancing functionality and reliability. Throughout the development cycle, rigorous testing and feedback mechanisms were employed to identify areas for improvement and implement necessary refinements. The iterative approach enabled continuous enhancement of the prototype, ensuring that each subsequent version addressed shortcomings identified in previous iterations.

Figure 6 : First prototype using thermocol

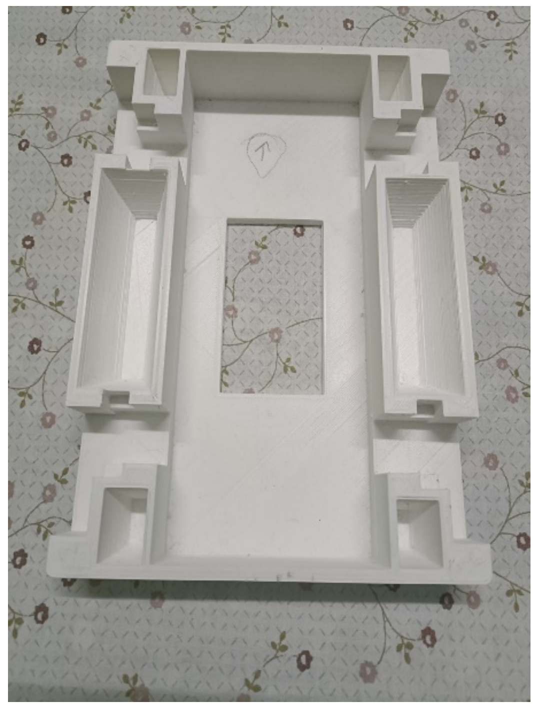


Figure 7 : 2nd Prototype , 3D printed

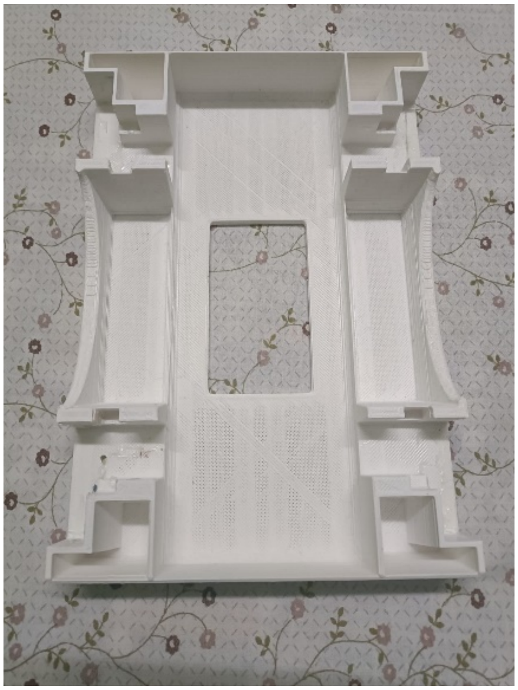


Figure 8: Improved 3D model

Figure 9: Further improvements

The prototype submission served as an opportunity to demonstrate the progress made in the development of Trashobot and discovering potential areas for further refinement and optimization.

3rd iteration of 3D model was 22% lighter while being 4% bigger in dimension compared to 1st iteration.

5.4 SUMMARY

Throughout the section, we discussed into the intricacies of Trashobot's hardware and software implementation, detailing the setup of essential components like the Raspberry Pi 4 and the development environment. The step-by-step coding process illuminates the evolution of Trashobot's functionalities, from basic car control to sophisticated robotic arm movements. The design and 3D printing of the robot car body and robotic arm components were also discussed. Challenges encountered during implementation, such as circuit design errors and motor calibration issues, are addressed along with corresponding solutions the Technical Implementation & Analysis section provides a comprehensive overview of Trashobot's development journey, highlighting achievements, and challenges faced during the project.

CHAPTER-6

PROJECT OUTCOME AND APPLICABILITY

6.1 KEY IMPLEMENTATION OUTLINES OF THE SYSTEM

Autonomous Navigation: Trashobot is equipped with advanced navigation capabilities, allowing it to traverse indoor environments autonomously. Utilizing a combination of sensors and mapping algorithms, the system can identify obstacles and chart optimal paths to navigate around them, ensuring efficient and safe operation.

Trash Detection and Classification: The heart of Trashobot lies in its ability to detect and classify various types of trash using computer vision algorithms. Integrated cameras capture visual data, which is then processed in real-time to identify objects of interest. Trash detection algorithms analyze the captured images, distinguishing between different types of waste and enabling targeted collection.

Precision Trash Picking: Trashobot's 6 DOF robotic arm enables precise and dexterous manipulation, allowing it to pick up trash with accuracy and efficiency. The robotic arm is capable of executing complex movement sequences to reach and grasp objects of various shapes and sizes, facilitating seamless trash removal without causing damage to surrounding objects or surfaces.

User-Friendly Interface: To enhance usability and accessibility, Trashobot features a user-friendly interface for remote control and monitoring. Users can interact with the system through intuitive interfaces, enabling them to supervise its operation, monitor trash collection progress, and intervene when necessary.

Scalability and Adaptability: Designed with scalability and adaptability in mind, Trashobot can be customized and configured to suit a wide range of environments and applications. Its modular architecture allows for easy integration of additional sensors, actuators, and functionalities, making it adaptable to evolving needs and requirements.

6.2 SIGNIFICANT PROJECT OUTCOMES

Efficient Waste Collection: Trashobot's autonomous navigation and trash detection capabilities enable it to efficiently collect and remove waste from indoor environments. By autonomously navigating through spaces and selectively picking up trash, Trashobot streamlines the waste collection process, reducing the time and effort required for manual cleaning.

Improved Waste Segregation: With its advanced computer vision algorithms, Trashobot can classify different types of waste in real-time, facilitating improved waste segregation. By accurately identifying and categorizing recyclable, organic, and non-recyclable waste, Trashobot contributes to more effective waste management practices and promotes recycling initiatives.

Enhanced Environmental Sustainability: By automating the process of waste collection and segregation, Trashobot helps reduce the environmental impact associated with traditional waste management methods. By promoting efficient resource utilization and reducing landfill waste, Trashobot contributes to a cleaner and more sustainable environment.

Increased Public Awareness: The deployment of Trashobot raises public awareness about the importance of waste management and the potential of robotics technology in addressing environmental challenges. By showcasing innovative solutions for waste collection and disposal, Trashobot inspires individuals and communities to adopt more responsible waste management practices.

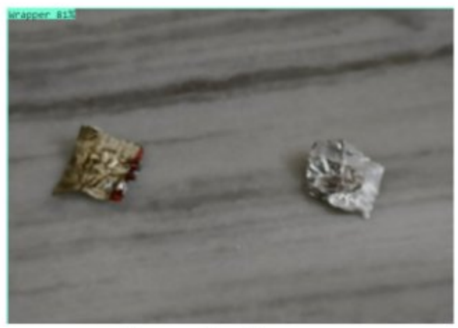
Future Research and Development Opportunities: The success of Trashobot opens up new avenues for future research and development in the field of robotic waste management. Areas for further exploration include the integration of advanced sensors for enhanced environmental monitoring, the development of AI-powered decision-making algorithms for adaptive waste collection, and the incorporation of IoT (Internet of Things) technologies for remote monitoring and control.

Sample-1

Sample- 2

Sample-3

Sample -4



Sample-5

Sample-6

Sample-7

Sample-8

Sample-9

Figure 10 :Examples of deformed household trash

Table 1: Accuracy results of object detection.

Sample	Object	Prediction	Level of confidence	Type of Categorized Trash
Sample 1	Slippers	Correct prediction	57% confidence	Not trash
Sample 2	Wrapper	Correct prediction	81% confidence	Trash
Sample 3	Bottle	Correct prediction	57% confidence	Not trash
Sample 4	Fruit	Wrong prediction	57% confidence	Trash
Sample 5	Wrapper	Correct prediction	81% confidence	Trash
Sample 6	Bottle	Correct prediction	66% confidence	Not trash
Sample 7	Water Bottle	Correct prediction	76% confidence	Not trash
Sample 8	Wrapper	Correct prediction	75% confidence	Trash
Sample 9	Bottle	Correct prediction	69% confidence	Not trash

6.3 PROJECT APPLICABILITY ON REAL-WORLD APPLICATIONS

Residential Waste Management: Trashobot's capabilities make it well-suited for residential waste management applications, where it can autonomously navigate homes and apartments, collecting and segregating waste efficiently. By relieving homeowners of the burden of manual waste disposal, Trashobot enhances convenience and promotes cleaner living environments.

Commercial and Industrial Settings:_Trashobot can also be deployed in commercial and industrial settings, such as offices, warehouses, and manufacturing facilities, to streamline waste collection and management processes. Its ability to navigate complex indoor environments and identify specific types of waste makes it valuable for maintaining cleanliness and hygiene in large-scale operations.

Public Spaces and Municipalities: Municipalities and public agencies can benefit from Trashobot's capabilities in maintaining cleanliness and sanitation in public spaces such as parks, streets, and sidewalks. By autonomously collecting litter and waste, Trashobot contributes to the beautification of urban areas and promotes civic pride.

Environmental Monitoring and Cleanup: Trashobot can be deployed in environmentally sensitive areas such as beaches, rivers, and forests to assist in environmental monitoring and cleanup efforts. Equipped with sensors and cameras, Trashobot can identify and collect debris and litter, helping to preserve natural ecosystems and protect wildlife.

Educational and Research Institutions: Trashobot serves as an educational tool for teaching students about robotics, artificial intelligence, and environmental sustainability. Educational institutions can use Trashobot to engage students in hands-on learning experiences, fostering interest and expertise in STEM (Science, Technology, Engineering, and Mathematics) fields.

Hospitality and Tourism Industry: Trashobot can enhance the cleanliness and aesthetic appeal of hotels, resorts, and tourist attractions by autonomously collecting waste and maintaining cleanliness standards. Its discreet operation and non-intrusive design make it suitable for deployment in customer-facing environments without disrupting guest experiences.

6.4 INFERENCE

We looked at the transformative potential of Trashobot in revolutionizing waste management practices and addressing environmental challenges. Through the development and implementation of Trashobot, significant outcomes have been achieved, including efficient waste collection, improved waste segregation, enhanced environmental sustainability, increased public awareness, and opportunities for future research and development. Moreover, the applicability of Trashobot extends to various real-world applications, including residential waste management, commercial and industrial settings, public spaces, environmental monitoring and cleanup efforts, educational institutions, and the hospitality and tourism industry. Overall, Trashobot represents a promising solution to the pressing challenges of waste management, offering innovative approaches to promoting cleanliness, sustainability, and efficiency in diverse settings.

CHAPTER-7

CONCLUSIONS AND RECOMMENDATION

7.1 OUTLINE

The key outcomes and achievements of this project include:

- Successful design and construction of Trashobot, comprising a robot car with a 6 DOF robotic arm.
- Integration of computer vision algorithms for autonomous navigation, trash detection, and classification.
- Demonstrated ability to navigate indoor environments, detect various types of trash, and execute precise picking motions.
- Developed a Machine learning algorithm which achieved an average accuracy of ~88.89% (identifying the trash accuracy), and the average confidence while correct prediction of trash is ~72.14% (Table 1).

7.2 LIMITATIONS/ CONSTRAINTS OF THE SYSTEM

Some of the limitations and constraints for the Trashobot could include:

- **Technical Complexity**: The design and implementation of Trashobot involve intricate mechanical and electronic components, leading to potential challenges in assembly, calibration, and troubleshooting.

- **Sensor Limitations**: The accuracy and reliability of sensors, such as cameras and proximity sensors, may be limited, affecting the precision of trash detection and navigation.

- **Power Consumption**: Continuous operation of motors and processing units may result in high energy consumption, requiring careful management of power resources to ensure sustained performance.

- **Environmental Factors**: Trashobot's performance may be affected by environmental conditions such as lighting, terrain, and obstacles, posing challenges to consistent operation in real-world settings.
- **Resource Constraints**: Limited availability of financial resources, materials, and technical expertise may constrain the scope and scale of the Trashobot project, impacting its development and deployment.

- **Time Constraints**: Strict project timelines and deadlines may limit the duration and depth of research, development, and testing phases, potentially compromising the thoroughness and completeness of the system.

- **Safety Considerations**: Safety risks associated with robotic operations, including collision hazards and mechanical failures, must be carefully managed to prevent damage to property and ensure user safety.

- **Regulatory Compliance**: Compliance with regulatory requirements and standards for robotics and waste management may pose additional constraints and complexities in the design and operation of Trashobot.

7.3 FUTURE ENHANCEMENTS

There are several possible future enhancements, such as:

- **Advanced Sensor Integration**: Incorporate state-of-the-art sensors, such as LiDAR and advanced cameras, to enhance the accuracy and reliability of trash detection and navigation capabilities.

- **Machine Learning Algorithms**: Implement machine learning algorithms for dynamic adaptation and optimization of Trashobot's behavior in response to changing environmental conditions and user preferences.

- **Multi-Robot Collaboration**: Explore the feasibility of deploying multiple Trashobot units in coordinated teams to achieve greater coverage and efficiency in waste collection and management tasks.

- **Autonomous Charging Station**: Develop an autonomous charging station for Trashobot to enable self-charging capabilities, extending operational uptime and reducing reliance on external power sources.

- **Environmental Monitoring**: Integrate environmental monitoring sensors to enable Trashobot to assess and report on environmental conditions, such as air quality and pollution levels, in addition to waste management tasks.

- **Smart Waste Sorting**: Enhance Trashobot's capabilities for waste sorting and segregation using advanced computer vision and robotic manipulation techniques to improve recycling efficiency and reduce landfill waste.

- **Remote Monitoring and Control**: Implement remote monitoring and control features to enable users to remotely supervise Trashobot's operations, receive real-time status updates, and intervene when necessary.

- **Modular Design Architecture:** Adopt a modular design architecture for Trashobot to facilitate easy upgrades, maintenance, and customization, enabling seamless integration of new technologies and functionalities as they become available.

- **Human-Robot Interaction:** Explore opportunities for enhancing human-robot interaction through intuitive interfaces, voice commands, and augmented reality displays to improve user experience and acceptance.

- **Sustainability Initiatives:** Align Trashobot's development roadmap with broader sustainability initiatives, such as circular economy principles and zero-waste strategies, to maximize its positive impact on environmental conservation and resource efficiency.

7.4 INFERENCE

The Trashobot project has demonstrated significant progress in the development of a robotic solution for waste management. Through innovative design, integration of advanced technologies, and rigorous testing, Trashobot has shown promise in addressing the challenges of waste collection and disposal. The implications of Trashobot's capabilities extend beyond waste management, with potential applications in various industries and sectors requiring automated solutions. Despite encountered limitations and challenges, valuable insights have been gained for future improvements and advancements. Moving forward, recommendations include further optimization of sensor accuracy, refinement of algorithms, and collaboration for continued innovation. With ongoing research and development efforts, Trashobot holds promise for a cleaner, more sustainable future, contributing to environmental conservation and resource efficiency.

APPENDIX A (Screenshots & Pictures)

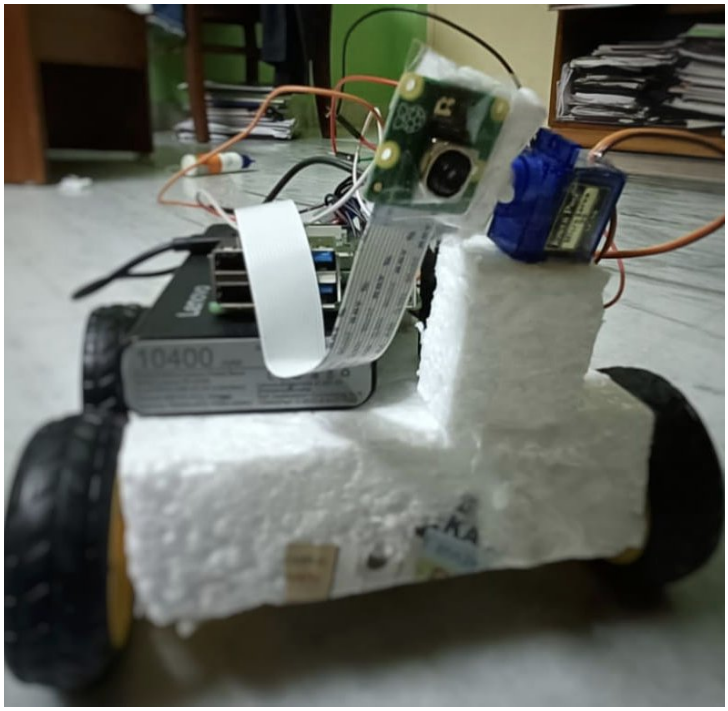


Figure : Robot car prototype

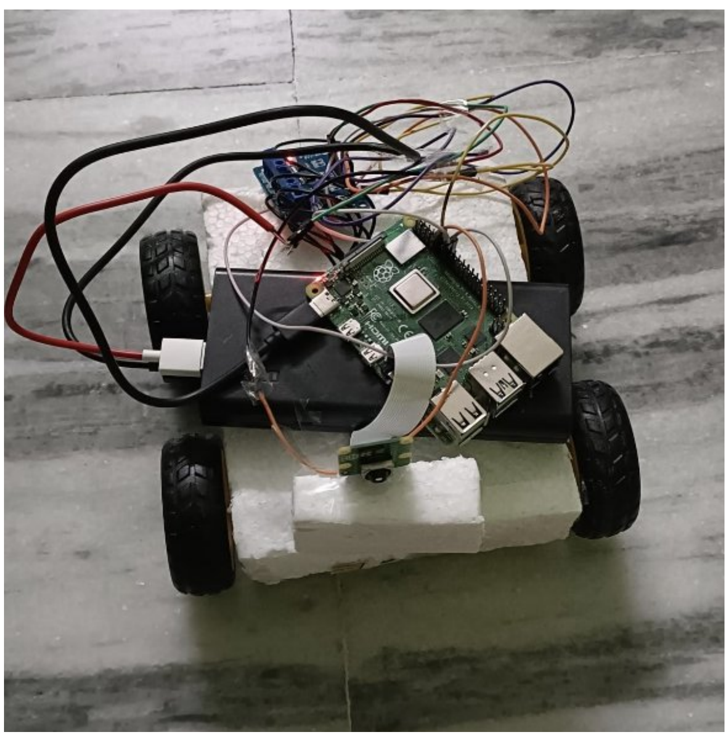


Figure: Robot car prototype

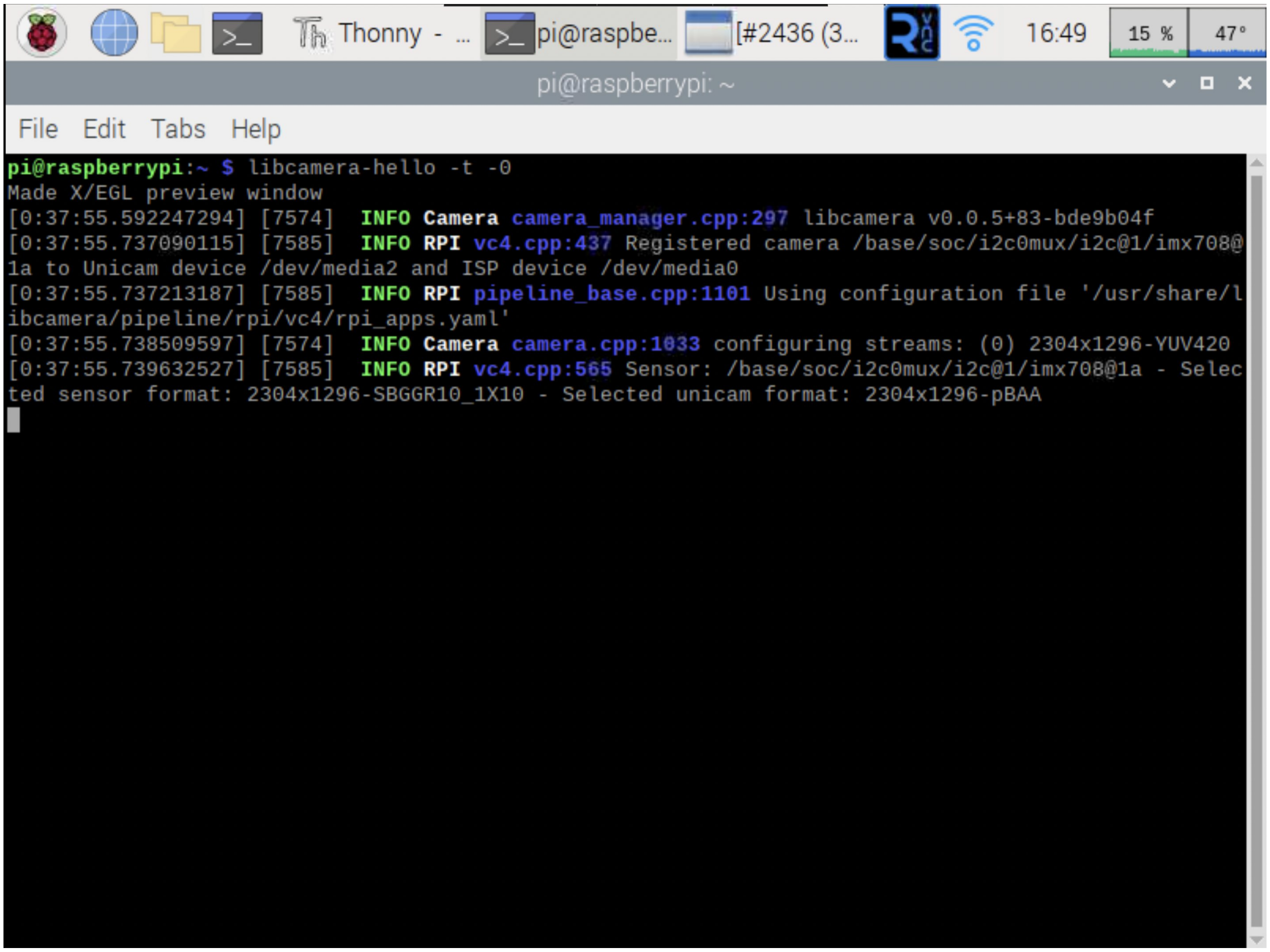


Fig: Screenshot of Raspberry pi terminal running camera operation commands

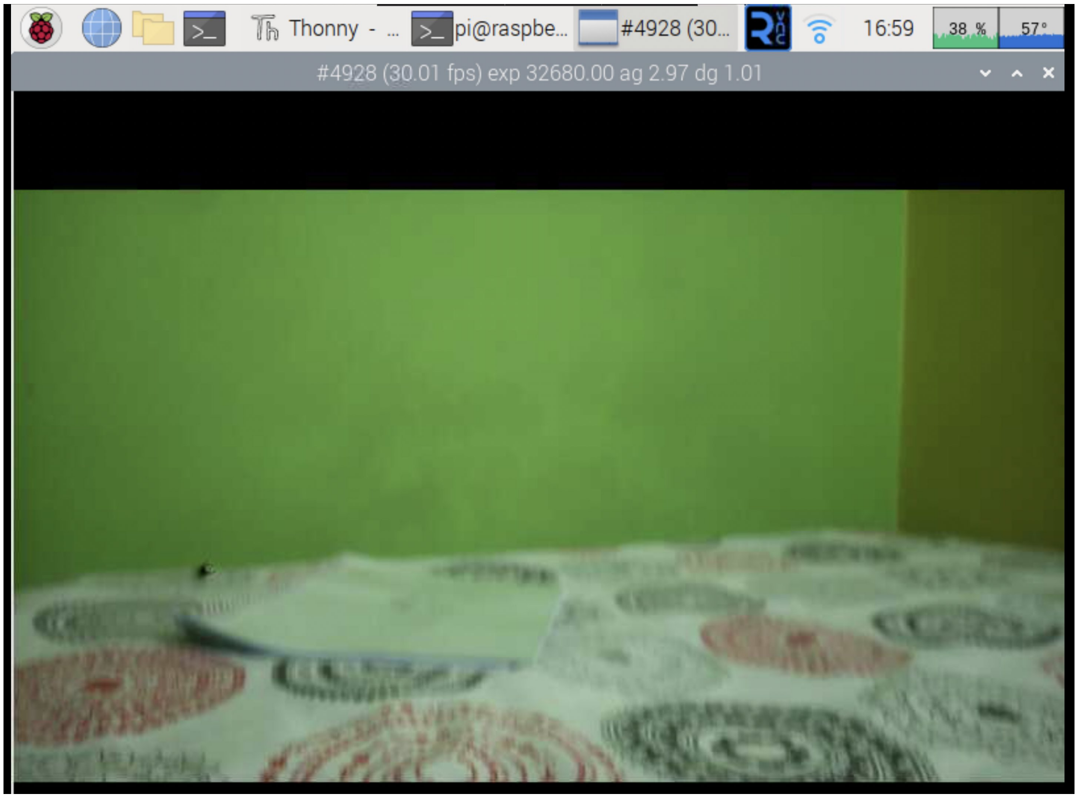


Figure: picture using raspberry pi camera module

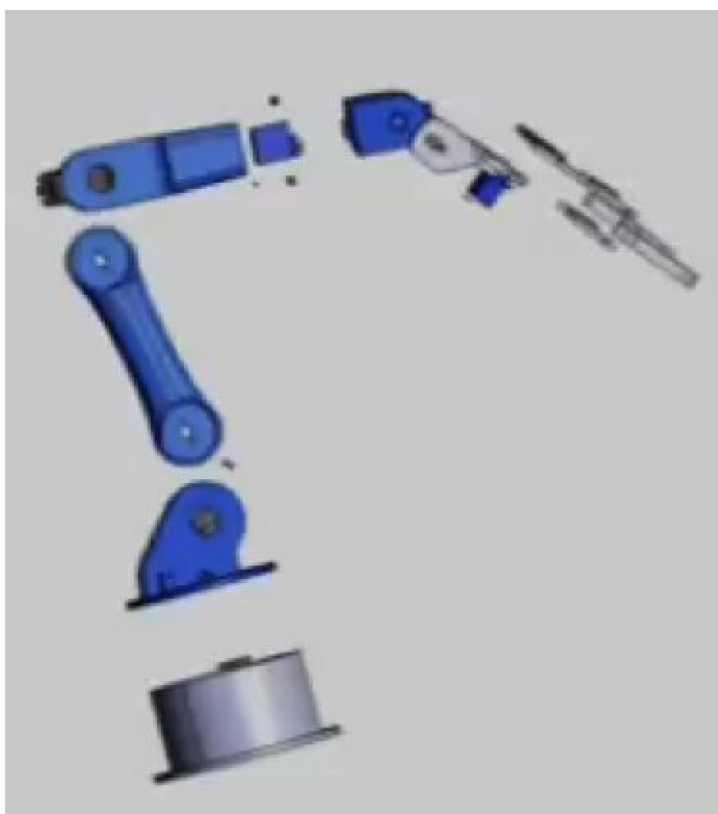


Figure: 3D design of robot arm

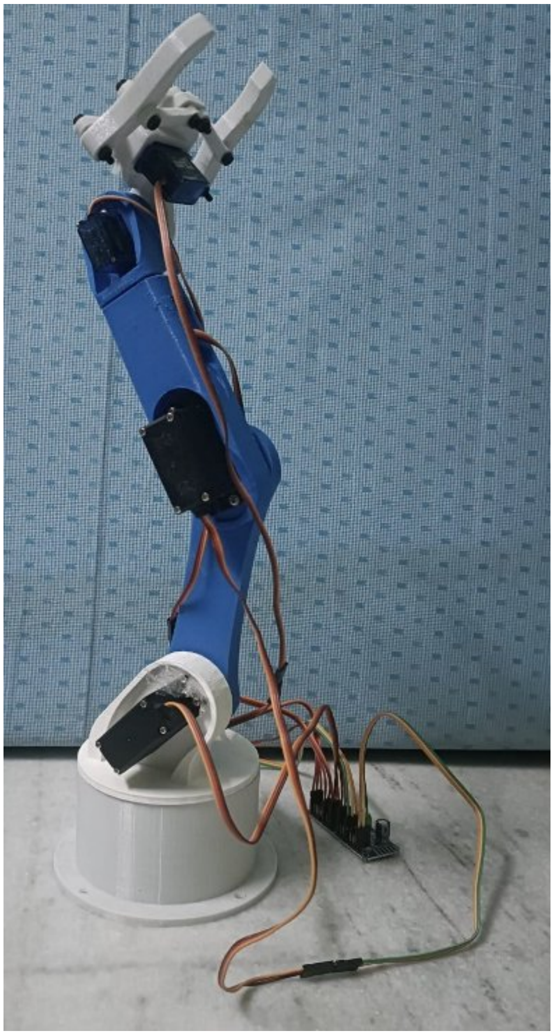


Figure: Robot arm full assembly

Figure : Full Trashobot assembly

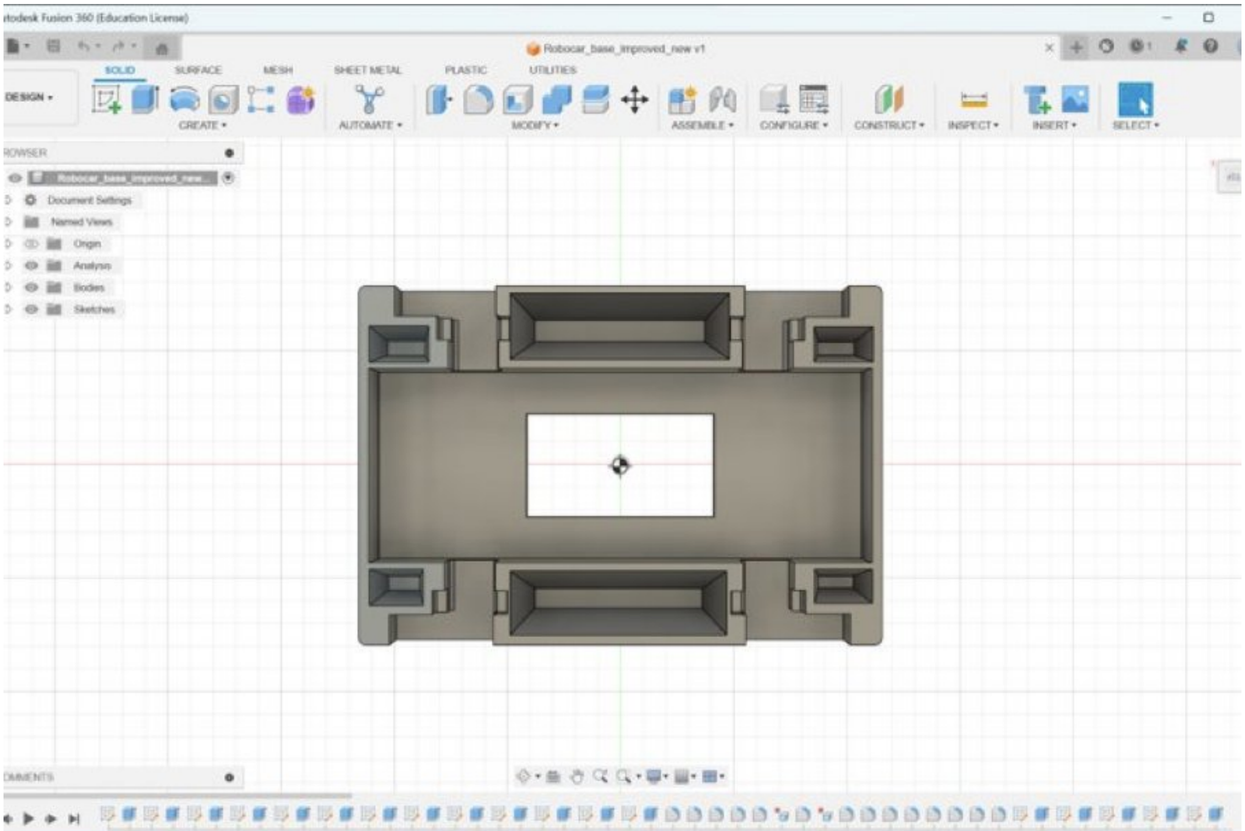


Figure: 3D model robot car base

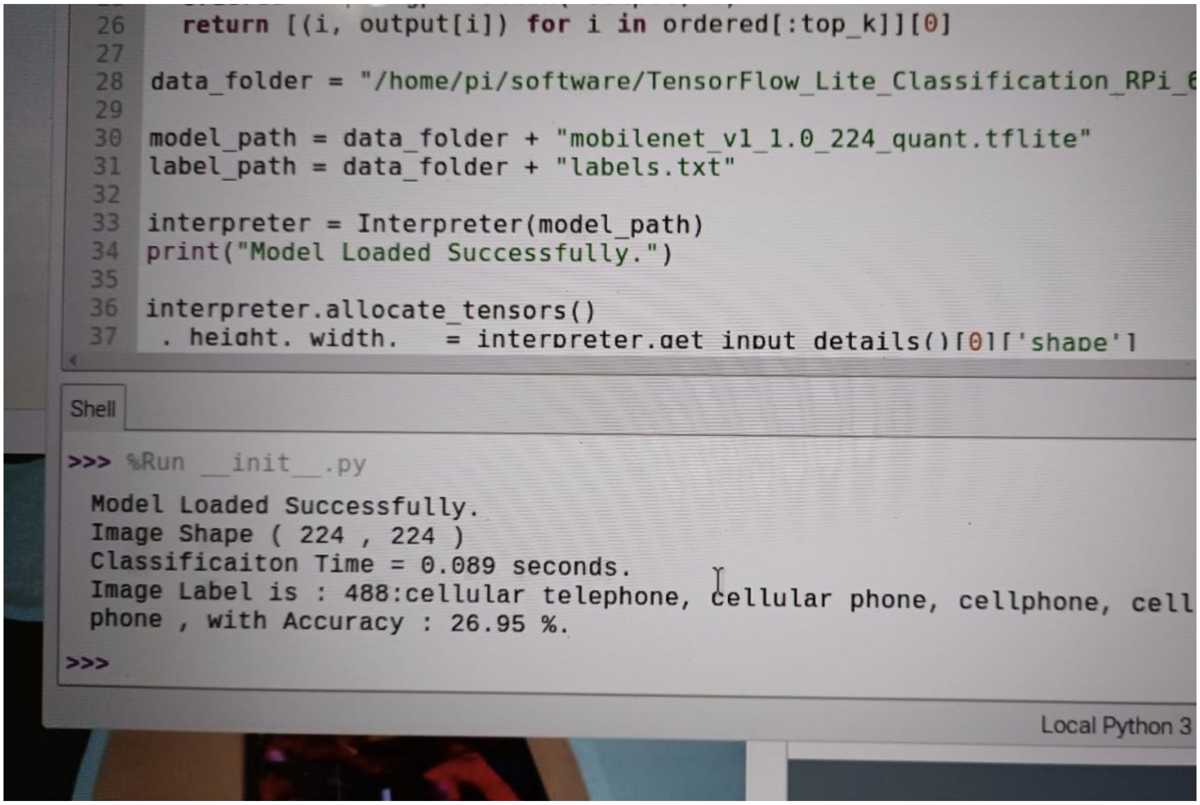


Figure: Object detection

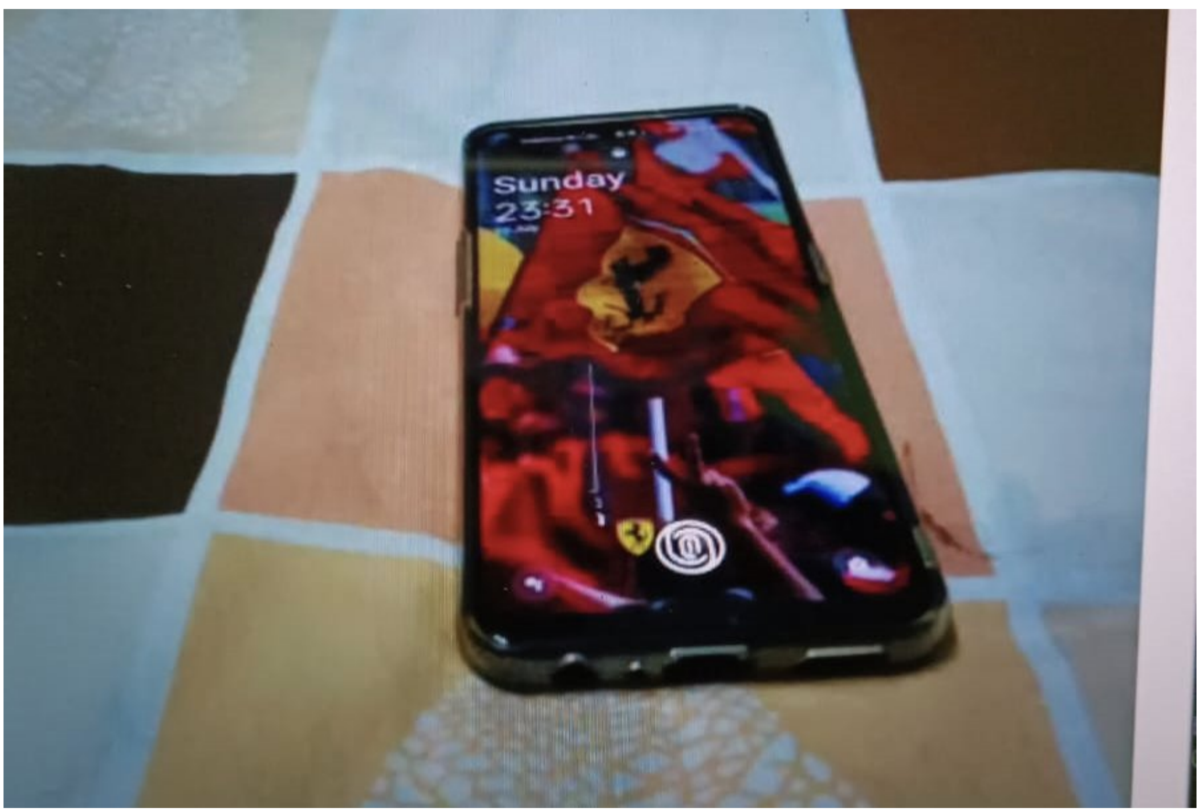


Figure: Image used in above object detection

Appendix-B (Coding)

Code for controlling robot car with 4 motors, servo motor attached camera and LED lights:

```
import RPi.GPIO as GPIO
import time
from pynput import keyboard

# Set GPIO mode
GPIO.setmode(GPIO.BCM)

# Define motor pins
motor1_pin1 = 17
motor1_pin2 = 18
motor2_pin1 = 22
motor2_pin2 = 23

# Define servo motor pin
servo_pin = 8

# Define LED (motor3) pin
motor3_pin1 = 24
motor3_pin2 = 25

# Set motor pins as output
GPIO.setup(motor1_pin1, GPIO.OUT)
GPIO.setup(motor1_pin2, GPIO.OUT)
GPIO.setup(motor2_pin1, GPIO.OUT)
GPIO.setup(motor2_pin2, GPIO.OUT)

# Set servo pin as output
GPIO.setup(8, GPIO.OUT)
```

```python
# Set LED (motor3) pins as output
GPIO.setup(motor3_pin1, GPIO.OUT)
GPIO.setup(motor3_pin2, GPIO.OUT)

# Create servo object
pwm=GPIO.PWM(8, 50)
pwm.start(0)

# Function to control the car's movements
def move_forward():
   GPIO.output(motor1_pin1, GPIO.HIGH)
   GPIO.output(motor1_pin2, GPIO.LOW)
   GPIO.output(motor2_pin1, GPIO.HIGH)
   GPIO.output(motor2_pin2, GPIO.LOW)

def move_backward():
   GPIO.output(motor1_pin1, GPIO.LOW)
   GPIO.output(motor1_pin2, GPIO.HIGH)
   GPIO.output(motor2_pin1, GPIO.LOW)
   GPIO.output(motor2_pin2, GPIO.HIGH)

def turn_left():
   GPIO.output(motor1_pin1, GPIO.HIGH)
   GPIO.output(motor1_pin2, GPIO.LOW)
   GPIO.output(motor2_pin1, GPIO.LOW)
   GPIO.output(motor2_pin2, GPIO.HIGH)

def turn_right():
   GPIO.output(motor1_pin1, GPIO.LOW)
   GPIO.output(motor1_pin2, GPIO.HIGH)
   GPIO.output(motor2_pin1, GPIO.HIGH)
   GPIO.output(motor2_pin2, GPIO.LOW)

def stop():
   GPIO.output(motor1_pin1, GPIO.LOW)
```

```
    GPIO.output(motor1_pin2, GPIO.LOW)
    GPIO.output(motor2_pin1, GPIO.LOW)
    GPIO.output(motor2_pin2, GPIO.LOW)

def toggle_led():
    GPIO.output(motor3_pin1, GPIO.LOW)
    GPIO.output(motor3_pin2, GPIO.HIGH)

def on_press(key):
    try:
        if key == keyboard.Key.up:
            move_forward()
        elif key == keyboard.Key.down:
            move_backward()
        elif key == keyboard.Key.left:
            turn_left()
        elif key == keyboard.Key.right:
            turn_right()
        elif key.char == 'w':
            move_servo(150)
        elif key.char == 's':
            move_servo(105)
        elif key.char == 'l':
            toggle_led()

    except AttributeError:
        pass

def on_release(key):
    if key in [keyboard.Key.up, keyboard.Key.down, keyboard.Key.left, keyboard.Key.right ]:
        stop()

def move_servo(angle):
    duty = angle / 18 + 2
    GPIO.output(8, True)
```

```
    pwm.ChangeDutyCycle(duty)
    time.sleep(1)
    GPIO.output(8, False)# Give the servo time to reach the position
    pwm.ChangeDutyCycle(0)
    #print("Moving servo to angle:", angle)

# Main program
listener = keyboard.Listener(on_press=on_press, on_release=on_release)
listener.start()

# Start servo PWM
pwm.start(0)

try:
    while True:
        time.sleep(0.1)

except KeyboardInterrupt:
    GPIO.cleanup()
    pwm.stop()
    listener.stop()
```

Code for Robot arm servo motor testing :

```
import RPi.GPIO as GPIO
from time import sleep

# Set the GPIO mode and pin numbers
GPIO.setmode(GPIO.BCM)
servo_pins = [16, 26]
```

```python
# Set the servo GPIO pins as outputs
for pin in servo_pins:
    GPIO.setup(pin, GPIO.OUT)

# Set up PWM for both servos
pwm = [GPIO.PWM(pin, 50) for pin in servo_pins]  # 50 Hz frequency

# Start PWM for both servos
for p in pwm:
    p.start(0)

try:
    while True:  # Run continuously
        for angle in range(30, 151):  # Move from 30 to 150 degrees
            for i, pin in enumerate(servo_pins):
                duty_cycle = angle / 18.0 + 2.5
                pwm[i].ChangeDutyCycle(duty_cycle)
            sleep(0.01)  # Small delay for smoother movement

        for angle in range(150, 29, -1):  # Move back from 150 to 30 degrees
            for i, pin in enumerate(servo_pins):
                duty_cycle = angle / 18.0 + 2.5
                pwm[i].ChangeDutyCycle(duty_cycle)
            sleep(0.01)

except KeyboardInterrupt:
    pass

# Cleanup when the program exits
for p in pwm:
    p.stop()
GPIO.cleanup()
```

Code for Object Detection using TensorFlow lite:

```
from tflite_runtime.interpreter import Interpreter
from PIL import Image
import numpy as np
import time

def load_labels(path):
  # Read the labels from the text file as a Python list.
  with open(path, 'r') as f:
    return [line.strip() for i, line in enumerate(f.readlines())]

def set_input_tensor(interpreter, image):
  tensor_index = interpreter.get_input_details()[0]['index']
  input_tensor = interpreter.tensor(tensor_index)()[0]
  input_tensor[:, :] = image

def classify_image(interpreter, image, top_k=1):
  set_input_tensor(interpreter, image)

  interpreter.invoke()
  output_details = interpreter.get_output_details()[0]
  output = np.squeeze(interpreter.get_tensor(output_details['index']))

  scale, zero_point = output_details['quantization']
  output = scale * (output - zero_point)

  ordered = np.argpartition(-output, 1)
  return [(i, output[i]) for i in ordered[:top_k]][0]

data_folder = "/home/pi/software/TensorFlow_Lite_Classification_RPi_64-bits/"

model_path = data_folder + "mobilenet_v1_1.0_224_quant.tflite"
label_path = data_folder + "labels.txt"
```

```
image_path = "/home/pi/image.jpg"  # Path to the saved image

interpreter = Interpreter(model_path)
print("Model Loaded Successfully.")

interpreter.allocate_tensors()
_, height, width, _ = interpreter.get_input_details()[0]['shape']
print("Image Shape (", width, ",", height, ")")

# Load the image saved previously using libcamera-jpeg.
image = Image.open(image_path).convert('RGB').resize((width, height))

# Classify the image.
time1 = time.time()
label_id, prob = classify_image(interpreter, image)
time2 = time.time()
classification_time = np.round(time2 - time1, 3)
print("Classification Time =", classification_time, "seconds.")

# Read class labels.
labels = load_labels(label_path)

# Return the classification label of the image.
classification_label = labels[label_id]
print("Image Label is:", classification_label, ", with Accuracy:", np.round(prob * 100, 2),
"%.")
```

REFERENCES

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. In TensorFlow Dev Summit.
2. Autodesk. (2023). Fusion 360 features. In Autodesk.
3. Barba-Gonzaga, P., Zamora-Castro, S. A., & Ortiz-Salinas, R. (2021). Smart bin for segregation and classification of municipal solid waste using artificial intelligence and internet of things. *Waste Management*, 136, 13-21.
4. Glouche, Y., & Couderc, P. (2013). A smart waste management with self-describing objects. In The Second International Conference on Smart Systems, Devices and Technologies (SMART'13).
5. Hugging Face. (2023). MobileNet v1 model documentation. In Hugging Face Transformers Documentation.
6. Lakshmi, V., Sathees, S., Rani, A., Rajasekaran, K., & Ravi, V. (2021). A review of trash collecting and cleaning robots. In 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO).
7. Matsumoto, K., & Lee, J. (2023). Development of smart garbage box using artificial intelligence and IOT: toward selective collection of garbage. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 27(4), 622-628.
8. Raspberry Pi Foundation. (2023). Raspberry Pi computers documentation. In RaspberryPi Documentation.
9. TensorFlow. (2023). Object detection with TensorFlow Lite. In TensorFlow.