A Project Report

On

EnviroBot

A Mobile Robotic Arm for Efficient Cleaning and Task Automation

---

Submitted For Partial Fulfillment of The Criteria For
Bachelor Of Science (Engineering)
In
Information & Communication Technology
By

Jewel Nath	(ID 11909037)
Tasmiara Jahan Toma	(ID 11909048)
Tanjina Akter	(ID 11809008)

SESSION: 2018-19

Under the Supervision of
Amena Begum
Assistant Professor

COMILLA UNIVERSITY

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY
FACULTY OF ENGINEERING,
COMILLA UNIVERSITY (KOTBARI, CUMILLA, BANGLADESH)

14 January, 2024

Letter of Transmittal

January 14, 2024
To
Amena Begum
Assistant Professor
Department of Information and Communication Technology
Comilla University

Subject: Project Report Submission - "EnviroBot: A Mobile Robotic Arm for Efficient Cleaning and Task Automation"

Dear Respected Amena Begum Mam,

We are pleased to submit the project report titled "EnviroBot: A Mobile Robotic Arm for Efficient Cleaning and Task Automation." This project was undertaken as part of our coursework in the Department of Information and Communication Technology at Comilla University.

Throughout the development of this project, we encountered various challenges that have significantly enriched our experiences and knowledge. The report aims to articulate our understanding and insights gained during the exploration of advanced AI methods for waste detection.

Despite the obstacles faced, we have diligently invested our efforts in creating a comprehensive and meaningful project report. We believe that our endeavor not only deepened our understanding of the subject matter but will also contribute positively to the academic discourse on waste detection using robotic arms.

We trust that the content of our project report will be informative and engaging. Your feedback and analysis are crucial to us, and we would be grateful for your insights.

We appreciate your guidance and support throughout this project. It has been a valuable learning experience under your supervision, and we eagerly await your feedback.

Yours sincerely,

Jewel Nath
Tasmiara Jahan Toma
Tanjina Akter
Department of Information and Communication Technology
Comilla University

Declaration

"We confirm that this submission is an authentic work, conforming to the guidelines and instructions set forth by our supervisor, sir. To the best of our knowledge and belief, it does not contain any material previously authored by another person that has been acknowledged or approved for the award of any other degree or diploma from the university or any other higher education institution, unless duly acknowledged within the text."

Jewel Nath
ID-11909037, Session (2018-19)
Department of Information and Communication Technology
Comilla University

Tasmiara Jahan Toma
ID-11909048, Session (2018-19)
Department of Information and Communication Technology
Comilla University

Tanjina Akter
ID-11809008, Session (2017-18)
Department of Information and Communication Technology
Comilla University

Certificate of Acceptance

This is to certify that the project report entitled **"EnviroBot: A Mobile Robotic Arm for Efficient Cleaning and Task Automation"** by Jewel Nath (ID 11909037), Tasmiara Jahan Toma (ID 11909048) & Tanjina Akter (ID 11809008) has been carried out as a partial fulfillment of requirement for the Degree of Bachelor of Science (Engineering) in Information and Communication Technology, Comilla University. The dissertation has been carried out under my guidance and is a record of the authentic work carried out successfully. Their performance has been satisfactory during this project period.

I wish their every success in life.

Supervisor:

....

Amena Begum

Assistant Professor

Department of Information and Communication Technology

Comilla University, Cumilla.

Acknowledgement

We wish to extend our heartfelt gratitude and appreciation to our esteemed project supervisor, Madam Amena Begum (Assistant Professor, Department of Information and Communication Technology at Comilla University), for her invaluable guidance, steadfast support, and insightful feedback throughout the entirety of this project. Her expertise and encouragement have played a pivotal role in shaping our understanding and approach.

We would like to convey our sincere thanks to each member of our team. The collaborative effort and unwavering dedication of every team member were crucial to the triumphant culmination of this project.

Furthermore, we acknowledge the support received from our peers, mentors, and the academic community. Their constructive inputs and engaging discussions have added depth to our project and broadened our perspectives.

In conclusion, we express our gratitude to our families and friends for their understanding, encouragement, and patience during the challenging phases of this project.

This project has been a journey of learning, growth, and collaboration, and we extend our sincere thanks to everyone who contributed to its successful completion.

Thanking you,
Jewel Nath
Tasmiara Jahan Toma
Tanjina Akter
Department Of Information and Communication Technology
Comilla University

ABSTRACT

Aim/Objectives: This project is dedicated to establishing an autonomous garbage collection and sorting system, aiming to segregate waste into categories such as paper, plastic, metal, and general trash. The primary objective is to design and develop a comprehensive solution to address waste disposal challenges in diverse environments, including schools, restaurants, offices, hotels, and similar locations. The process involves creating a robust solution, conducting thorough tests for effectiveness, and ensuring the ongoing maintenance of the implemented system.

Method: Our project consists of two primary components. Firstly, an automated system is under development for trash collection along a designated path. The system autonomously surveys its surroundings using a camera and employs the YOLO Detection Model, running on a Raspberry Pi. With ROS2 MoveIt, appropriate commands are sent to the Arduino, which then moves the robotic arm to collect and deposit waste into an attached bin. Additionally, a user-friendly software interface facilitates manual control, allowing commands to be sent to the robotic arm for collecting stationary waste. Furthermore, the system incorporates voice command functionality, enabling the robot to respond and collect trash based on verbal commands.

Findings: The implemented autonomous robotic arm for trash collection showcases versatile functionality. Capable of efficiently segregating and depositing trash into an attached bin, the robot features a camera for remote monitoring of garbage collection activities by administrators. Moreover, the prototype incorporates an electronic mechanism, allowing the robot to dispense the collected garbage at specified locations. Operated exclusively by installed batteries, the robot functions without the need for fuel or electricity.

LIST OF CONTENTS

LIST OF ABBREVIATIONS

Abbreviations	Full Form
HRI	Human-Robot Interaction
FPGA	Field Programmable Gate Array
LED	Light Emitting Diode
DT	Detection time
UART	Universal Asynchronous Receiver-Transmitter
DOF	Degree of Freedom
ROS	Robot Operating System
RVIZ	Robot Operating System Visualization
MWM	Mobilization With Movement
ICSP	In-circuit serial programming
IDE	Integrated Development Environment
IOT	internet of things
PET	Polyethylene Terephthalate
SoC	system-on-a-chip
URDF	unified robotics description format
YOLO	You Only Look Once

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Nowadays Robot technology plays an important role in industry and life. The use of robotic arms is very wide. In life, the robotic arm can be used for cooking, cleaning, and other household service tasks. The term trash is only applicable when the owner classifies the substance as rubbish

The average person produces over 2.5 kgs of solid waste every day. That adds up to over 1 tonne of trash annually for each global citizen. With the world's population exceeding 7 billion, effective waste management strategies are crucial to handle this enormous volume in an environmentally sustainable manner. Currently, most municipal systems rely on manual sorting and collection - a process that is labor intensive and fails to achieve optimal diversion rates.

How might we leverage technological innovations to address this growing global challenge? One promising approach utilizes machine learning and computer vision. By training algorithms to visually identify and categorize different materials, could an automated system sort through mixed waste streams with greater accuracy and efficiency than human workers? This study aims to develop a prototype of such a machine learning-powered trash collector and evaluate its performance for municipal solid waste applications.

Any robotic arm needs to have mobility to collect trash. Mobility Systems are robots aimed at transporting over variable distance payload. In. The arm contains multiple joints that act as axes that enable a degree of movement. The higher the number of rotary joints a robotic arm features, the more freedom of movement it has. In reality, Mobile robots are faced with unstructured environments, thus they have their own good mobile performance and obstacle-climbing ability, which means that they must have capable locomotion mechanisms. Mobile robots have various kinetic structural forms including wheeled, legged, tracked, and combinational mobile mechanisms.[1] For trash collected by robotic arm, we need a basement. The wheel and motor driver will be in the basement. It can be moved anywhere with the help of a motor.

Waste management is the process through which wastes are gathered, moved, and best method of disposal to reduce or eliminate their negative effects. Typical method of disposal is garbage collection, which involves regular pickup of rubbish by vehicles. The most expensive part of waste management is frequent trash collection and related transportation; however, technology is widely accessible to improve efficiency. Using a geographic information system as a starting point, a city may improve routing and reduce unfavorable truck usage.

Sensors can streamline routes and cut down on pointless pickups. By determining the amount of garbage in various containers, trash collection routes may be made more efficient.

The waste sorting and collection process can be conceptualized as a series of discrete steps that must be executed successfully by the machine learning-powered robotic system. These include: waste detection, rover navigation, material classification, grasping methodology, transport, and unloading into designated bins.

Waste detection is a critical initial phase that requires application of computer vision and deep learning techniques. Once a waste object is identified, the robot must autonomously navigate to its position for retrieval. This involves processing sensor feedback through a path planning algorithm to map efficient trajectories for the multi-jointed robotic arm. Precision motor control is then needed to maneuver each joint and effectively reach the targeted coordinate space. Classification of detected waste items into predefined typology (e.g. plastic, paper, organic) allows the system to select the appropriate grasping methodology.

Grasping and lifting of classified items from the ground or other surfaces requires dexterous manipulation of the end effector. Force sensing, iterative planning and tactile feedback help achieve a stable and secure grip during manipulation and transport. This process minimizes risk of spills or jostling as the waste is moved.

Once grasped, the robotic arm must safely transport the item to its designated drop-off point without collisions. Trajectory planning and real-time motion control ensure this is achieved efficiently while avoiding obstacles. Finally, the object is unloaded through precise articulation of the end effector into the corresponding collection bin.

The robotic rover is a small vehicle that can move over rough areas where it is tough for people to walk. NASA used it initially for space exploration which is known as Mars rover. A robotic rover feels no threat to personnel involved in life-threatening conditions or scenarios that human society needs to deal with. It can be considered to be made use of assisting security authorities in carrying out search and rescue operations besides gathering information and insights. However, It works very well in collecting trash in inaccessible places.

A camera in a robotic arm is used to capture visual information about the environment and the objects that the arm interacts with. The captured data is processed in what is known as image processing to extract information that needs to know about the object such as size, position and shape. The robotic arm uses this information as a guide to adjust its movements.

1.2 Objectives

Our system, EnviroBot, is meticulously designed to diminish human intervention, elevate waste disposal practices, and actively contribute to a more sustainable environment. Through process automation, we aim to enhance waste management efficiency and cleanliness. EnviroBot incorporates cutting-edge technologies, leveraging a combination of Raspberry Pi, Arduino, servo channels, servos, motor drivers, motors, YOLO detection model, and LiPo batteries. Our specific objectives include:

1. **Utilizing Deep Learning Techniques and Robotic Framework:**
   - Employing deep learning techniques, facilitated by the YOLO detection model, to identify diverse types of waste with precision.
   - Integrating the ROS2 framework for seamless communication between components, and MoveIt to calculate wheel and servo movements for efficient navigation to garbage locations, precise garbage collection, and placement into an attached bin.
2. **Data Collection and Storage:**
   - Systematically storing collected data, timestamped by date, to track waste trends over time.
   - Harnessing this data to enhance waste management practices and devise innovative strategies for waste reduction.
3. **Reduction of Human Intervention using Robotic Components:**
   - Designing EnviroBot to significantly reduce the need for human involvement in waste management.
   - Utilizing Raspberry Pi, Arduino, servo channels, servos, motor drivers, motors, and LiPo batteries to automate the entire waste collection and disposal process.
4. **Exploration of Predictive Modeling:**
   - Exploring the potential of machine learning algorithms in conjunction with robotic components to predict waste trends.
   - Analyzing the collected data to identify patterns and trends, and subsequently developing predictive models for optimizing waste management practices.

By incorporating state-of-the-art technology and a comprehensive understanding of waste management processes, EnviroBot aims to revolutionize waste disposal practices, making significant strides towards a cleaner and more sustainable environment.

Here:

- Chapter 1 provides a brief introduction to our project,
- Chapter 2 explains the literature review done in waste management by Robotic arm.
- Chapter 3 describes the methodology used in this project.
- Chapter 4 shows and explains the results achieved.
- Chapter 5 concludes the research and describes future work.

CHAPTER 2

LITERATURE REVIEW

2.1 LITERATURE REVIEW

Urban areas face a constant struggle in efficiently collecting and disposing of waste. Traditional methods involve manual labor, which is not only resource-intensive but also prone to inefficiencies, delays, and often results in incomplete waste collection. As cities continue to expand, there is a growing need for innovative solutions that can enhance the effectiveness of waste management systems.

The integration of robotics and automation technologies presents an opportunity to address these challenges effectively. Our project focuses on developing a sophisticated yet practical solution that employs a robotic arm for automated trash collection. This technology not only streamlines the waste collection process but also contributes to a cleaner and healthier urban environment.

George Nantzios et al presents the design and implementation of a low-cost robotic arm assistant with voice interaction using machine vision. The system, based on the Niryo-One robotic arm, has been modified to detect and deliver objects with high accuracy and includes a Human-Robot Interaction (HRI) voice-controlled software for ease of use. The aim is to provide a cost-effective robotic assistant for everyday consumers, especially those with physical and mobility impairments. The system utilizes tools and techniques such as shape and color detection, the YOLO algorithm for object recognition, and QR code expandability. Experimental results show a 96.66% accuracy in detecting and delivering objects. Additionally, the system incorporates cloud processing and Field Programmable Gate Array (FPGA) boards for scalability and cost-effectiveness. The paper also discusses future enhancements, including adding LED lights, incorporating sensors for object detection, and improving the user interface. [2]

The suggested system by Muhammad Abbas Khan et al presents the design and development of a semi-autonomous garbage collector robot intended for efficient waste collection and disposal in diverse settings. Equipped with a robotic arm, remote monitoring camera, and controllable via voice command or a mobile application, the robot addresses waste management issues in Pakistan with the goal of reducing environmental pollution. The article also discusses the implementation method, flow diagram, and project results. Additionally, it highlights the broader context of other research papers and projects focused on automated garbage collection using robots and object detection technology, all aiming to enhance waste management systems. [3]

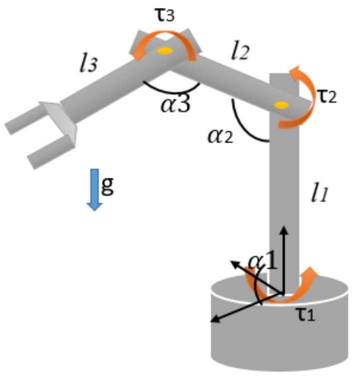
The paper discusses the development of an autonomous garbage collector robot called RoboDumpster, which utilizes a robotic arm with infrared sensors for obstacle avoidance and can locate, pick up, and empty garbage cans. The design and fabrication process of the robot, including the robotic arm, is described. The robot is a response to the Swachh Bharat Abhiyan initiative and aims to reduce the need for human labor in garbage collection. Additionally, the article highlights the advantages of robotic arms, such as efficiency, error-free operation, and assistance for the elderly and children with disabilities. The conclusion emphasizes the development of a 7 DOF autonomous robotic arm and its compatibility with various programming languages, and references related research and development in the field of robotics .[4]

Medhasvi Kulshreshtha et al [5] discusses the development and implementation of autonomous cleaning robots and garbage collection robots, focusing on their design, components, and operational capabilities. It covers topics such as deep learning-based object detection for robot navigation, path planning, and human-robot interaction. The paper also compares the performance of different computer vision models for object detection and highlights the use of the YOLOv4-tiny machine learning algorithm for trash detection in autonomous robots. Additionally, it addresses the challenges in implementation, robot operation, and simulation on various terrains, showcasing the robots' ability to overcome obstacles and collect trash. The study emphasizes the cost-effectiveness, multi-terrain capability, and simultaneous multi-object pickup mechanism of the autonomous trash-collecting robots.

Mask-RCNN achieved an average precision (mAP) of over 83% , YOLOv4 achieved 97.1% and YOLOv4-tiny achieved 95.2% and detection time (DT) respectably 3973.29 ms, 32.76 ms DT and 5.21 ms DT. As YOLOv4-tiny is very similar mAP to YOLOv4, but a much lower DT, so that YOLOv4-tiny was selected .[6]

P.Sunitha Devi et al. discussed the paper "Robotic ARM for Effective Environmental Cleaning " is about the development of a robotic arm for environmental cleaning, controlled remotely from a central hub. It emphasizes the use of UART communication protocol, Raspberry Pi, and servo motors in the assembly of the arm. The potential for reducing risk to human workers and improving efficiency in hazardous environments is highlighted, with future scope for automation using image processing and machine learning algorithms. The paper also discusses the advantages of robotic arms in operating in dangerous environments and places that cannot be reached by humans. Additionally, it mentions the potential for robotic arms to handle difficult and complicated operations faster and more accurately, as well as their implementation in vulnerable environments and free time activities. [7]

Vinodha et al. implemented a solution for manually managing waste and monitoring bins in their study. Their bin is equipped with Raspberry-Pi integrated with a camera for object detection and a sensor for level detection. They have used the YOLO algorithm for object detection. Their smart bin is connected to a mobile app for monitoring bins.[8]



There are several types of robotic arm . Those robotic arms are different by degree of freedom(dof). Hussain et al. presents the design and analysis of a 3 degree of freedom robotic arm, covering kinematic and dynamic analysis, as well as trajectory planning. It demonstrates agreement between analytical and numerical results, affirming the suitability of the robotic arm design for control system implementation. Additionally, the paper explores the use of software for simulations and compares the results with analytical calculations.[9]

Mohammed et al. choice of using a 4 degree-of-freedom (4-DOF) robotic arm in the paper is likely due to the specific application or research focus. The authors may have selected a 4-DOF robotic arm to demonstrate kinematics modeling and simulation in a simplified yet representative manner. This choice allows for a clear and manageable illustration of the kinematic analysis and simulation procedures without the complexity of higher DOF systems. Additionally, a 4-DOF robotic arm can still exhibit a wide range of motion and perform various tasks, making it suitable for educational and research purposes.[10]

Megalingam et al. choice of a 6 DOF robotic arm in this paper is based on the need for a greater work envelope, which is beneficial for industrial and research purposes . Additionally, the first 3 DOF provide positional coordinates for the end-effector, while the last 3 DOF control the orientation of the end-effector . This allows for a wide range of motion and flexibility in controlling the arm, making it suitable for various applications such as search and rescue operations in disaster-hit areas.[11]

The human arm has six degrees of freedom, making a 6-DOF robotic arm more capable of mimicking natural movements.The use of a 6-DOF robotic arm is driven by its ability to handle a wide range of tasks, providing flexibility, precision, and versatility in various applications across industries.3-DOF arms and 4-DOF robotic arms may face challenges in handling tasks that demand a high level of spatial flexibility and complexity.

Chikurtev et al. discusses the use of ROS, RVIZ, and MoveIt! for controlling a robotic arm manipulator. It describes the software and methods used, as well as the specifications of the Mover4 robot. The paper also presents experiments and results, showing successful control of the robotic arm using ROS. The article discusses the use of Robot Operating System (ROS) for controlling a robotic manipulator. The goal is to achieve control over the manipulator's simulation model in the RVIZ environment. The software described in the article is based on ROS, RVIZ, and MoveIt! The experiments conducted verify the distance from the robot's hand and explore various methods of controlling the robotic arm when performing specific trajectories. Key terms include robotic arm, ROS, RVIZ, MoveIt!, kinematic model, control, and motion planning.[12]



Brian Raafiu et al. discussed about Four Wheel Mobile Robotic.Four Wheel Mobile Robotic is a choice with a variety of the functions in the industry and the application of the other, reliability and intelligence system of wheeled mobile robot become an option on a 4.0 generation industry.[15] Stabilization of four-wheel mobile robot is an important case for the system control of the mobile robotA 4-wheel rover is commonly used in robotic arms to provide mobility and stability. The rover allows the robotic arm to move and manipulate objects in different locations, making it versatile for various tasks. The combination of wheels provides efficient navigation, making it easier for the robotic arm to reach its target positions and perform tasks with precision.Four wheeled robots rarely lose stability while moving.


R.Čermák discusses in his paper the utilization of MATLAB tools and a webcam to control a Mitsubishi Melfa robotic arm, emphasizing the use of low-cost solutions and open-source tools for educational purposes. It covers the hardware and software employed, image capture and processing, trajectory planning, and communication with the robotic arm.[13]

Wandi Susanto et al. presents research on 3D object detection using multiple Kinect cameras, focusing on the use of point clouds and depth information to enhance object detection and classification in 3D space. The authors introduce a new dataset recorded with four Kinects to evaluate their approach and discuss related work, their method for object classification and detection, and the advantages of employing multiple cameras for 3D object detection. Additionally, they propose different strategies for integrating information from multiple cameras to enhance object detection. The paper also addresses the utilization of color and depth information from multiple Kinect cameras for 3D object detection, along with experiments and evaluations demonstrating the benefits of this approach. Furthermore, the authors highlight the challenges and future directions in this research area.[14]

For the focusing on the use of point clouds and depth information to enhance object detection and classification in 3D space we use Kinect camera which is not gained from the webcam.

VLADIMIR TADIC et al. discusses the application of Intel RealSense depth cameras in robotics for detecting and extracting obstacles based on depth information. It introduces the technology and working principle of RealSense cameras and presents experimental results demonstrating their capabilities in obstacle detection on a wall. The paper compares the D415 and D435 models of RealSense cameras, highlighting the differences and their suitability for different robotics applications. It concludes that the D415 model is more suitable for robotics and mentions that the research was funded by the European Union.[16]

Bobulski and Mariusz provided a method for sorting plastic garbage. The waste classifications that were recommended for categorization include high-density polyethylene, polypropylene, polystyrene, and polyethylene terephthalate. The findings demonstrated that automating trash sorting using processing of images and Artificial Intelligence allowed the development of practical systems that can be used in the real world. They mostly used plastic materials to achieve 99% accuracy.[17]

Othman et al. demonstrated an improved method for gathering object identification and measurements from video streams. They proposed a technique to measure objects in real-time video that makes use of the OpenCV libraries and incorporates smart edge detection, dilatation, and erosion algorithms. In the suggested approach, they preprocessed their picture. To improve speed and accuracy, the camera captured a frame and turned it into grayscale. The object-finding technique used a clever edge detector. A single object or a collection of related things may be 12 located using it. The revised, processed image was sent via a clever edge detector. The whole image was examined by the clever edge algorithm. The remaining holes in the outer frame were then filled using the erosion and dilation method. The innovative edge recognition technique used in this work makes use of morphological processes like a combination of erosion and dilation to improve performance and increase item detection accuracy. The photographs were precisely captured by the camera. Four tasks were carried out by the system under evaluation: recording frames, detecting edges, identifying objects, and figuring out the sizes of each item. The proposed method almost calculated the sizes of the objects properly, with a 98% success rate. [18]

Wang et al. looked at seven different CNN designs, including MobileNetV2 and MobileNetV3, ResNet50, ResNet101, ResNet152, Xception, and InceptionV3. Along with the TrashNet dataset, there were 17,073 images in all, showing nine different types of rubbish. A mobile app for smartphones was developed to gather garbage images, classify them, and send them to an application server for categorization. The interaction with mobile phone apps was made simple 8 by the use of a Bluetooth module. The Jetson Nano and the Flask web framework were used to build the classification approach. OneNET (an open IoT platform) received all the data for analysis and decision-making. They employed the MobileNetV3 classification algorithm of the MWM system and found that it had excellent accuracy in classification (94.24%) with a short execution time (261.7 ms), and low model complexity (49.5 MB). [19]

N.Genc et al. discussed about the CONTROL OF ARM ROBOTIC GUIDED BY GPS SYSTEM. Using the GPS and compass data plus the bearing and distance calculations which are based on the observed GPS readings, must be sufficient to direct the car with high accuracy toward any given point on the map. A typical explorer GPS location-based robot is described as a compact of autonomous remote controlled wireless object such as car, drone or rover, with the ability of auto guiding itself toward any given-on map coordinates.[20]

Chuet al. presented a MHS in their work for automatically categorizing garbage that people left behind in public spaces. The MHS used a CNN-based approach for gaining image characteristics with an MLP technique to aggregate image attributes along with other feature data to decide whether or not waste was recyclable. High-resolution cameras were used to take trash images, and sensors were used to find other important feature data. Manually labeled objects were used for training and testing the MHS, and in two different testing situations, it surpassed a benchmark CNN-based technique that simply considered visual inputs by reaching an overall classification accuracy of more than 90%. [21]

Bobulski and Kubanek employed deep learning and CNNs to assess different forms of plastic trash and segregate different material types in a group, with the possibility to only recycle particular types of plastic. To determine the type of plastic used to create the trash, they advised utilizing a machine with a CPU dedicated to image processing. The system classified plastic garbage using an RGB camera, a CPU, and computer vision software. Given that PET products are the most often recycled household waste, the majority of the pictures in the WaDaBa database, which stores the images used, are of PET goods. Their method also had the advantage of accelerating network learning. Their accuracy of 97.43% after 4 epochs was decent. Another benefit of using it for mobile devices like the Raspberry Pi platform was that, according to this research, it had fewer characteristics than other networks. [22]

GPS is used in robotic arms to accurately locate and track their position, enable autonomous navigation, and assist in planning their paths. It helps robotic arms operate efficiently in outdoor environments, large areas, or remote locations.

CHAPTER 3

METHODOLOGY

A brief description of the methods used in this project is given in this section. The developed technique in this project involved gathering the data, pre-processing the data, object detection techniques, monitoring bin level, and system integration.

3.1 Tools

Some hardware and software equipment have been used for this system.

The software used for our project are given below:

- Arduino IDE
- Arduino Programming
- ROS2 Humble
- Ubuntu OS 22.04
- Rviz
- Gazebo
- Xacro
- URDF
- Google Colab
- Python 3.10
- Machine Learning
- Jupyter Notebook
- VS studio

Arduino IDE: The Arduino Uno, powered by the ATmega328P, offers 14 digital I/O pins (6 PWM-enabled), 6 analog inputs, a 16 MHz ceramic resonator, USB connection, power jack, ICSP header, and a reset button. The Arduino IDE provides a user-friendly environment with a code editor, message area, console, and toolbar for program development. It facilitates program uploading and communication with Arduino hardware. Exclusive servo control is achieved through Arduino IDE programming. These compact boards serve as the brains for robots, enabling seamless coordination of various electronic components like buttons, motors, switches, and lights.

Arduino Programming: The Arduino programming language is harnessed to instruct microcontroller boards like the Arduino Uno, enabling seamless interaction with an array of sensors, actuators, and connected devices. Widely adopted in projects spanning robotics, home automation, and Internet of Things (IoT) applications, it serves as a versatile tool for bringing creative ideas to life.

ROS2 Humble: The Robot Operating System (ROS) comprises a suite of software libraries and tools crafted for constructing robot applications. Originating in 2007, ROS has witnessed significant advancements within the robotics community. ROS 2, introduced in 2016, stands as the enhanced iteration, evolving dynamically ever since. The ROS 2 project aims to embrace transformations in robotics, preserving the strengths of ROS 1 while enhancing its shortcomings. Notable improvements include addressing challenges in scalability, performance, security, and cross-platform compatibility.

Ubuntu OS 22.04: Canonical Ubuntu 22.04 LTS is the latest long-term support release in the globally acclaimed Ubuntu Linux distribution. It features Ubuntu Core, an efficient and secure operating system for connected devices, with system requirements including a 2 GHz dual-core processor or better, 4 GB of RAM, 25 GB of free hard drive space, and recommended internet access.

Ubuntu has consistently supported critical open-source projects in robotics for over a decade, contributing to the success of initiatives like ROS, PX4, Autoware, OpenCV, PCL, and more. The development process benefits from Ubuntu's responsive, user-friendly, regularly updated, lightweight, and secure nature.

Rviz: Rviz serves as a 3D visualization platform within ROS. It enables graphical representation of external information and facilitates the transmission of control instructions to objects through Rviz. This dual functionality allows for both the visual display of information and the monitoring and control of a robot.

Gazebo: Gazebo stands as an open-source 3D robotics simulator that faithfully replicates real-world physics within a high-fidelity simulation. This platform proves invaluable for developers, allowing them to swiftly test algorithms and craft robot designs within virtual landscapes. Serving as the simulation environment, Gazebo delivers accurate physics simulation and visual representation for robot arm testing and visualization.

Xacro: XACRO, a macro language, extends URDF to offer a more straightforward and reusable approach to describing robots. Introducing the concept of macros, XACRO provides reusable and parameterized components for robot descriptions, enhancing simplicity and efficiency in the process.

URDF: URDF (Unified Robot Description Format) stands as a specialized file format employed to intricately define the composition and arrangement of robots within the ROS (Robot Operating System) environment.It is the foundation for robot description in ROS and is required for a variety of applications such as robot visualization, motion planning, and simulation.

Python3: Python stands as a versatile and potent general-purpose language, offering myriad applications. The latest iteration, Python 3, caters to both novice and experienced developers, establishing itself as a top choice in the realm of programming languages worldwide.

Google Colab: Google Colab presents a cost-free, cloud-centric platform where users can collaboratively write and execute Python code within a seamless environment. It is not explicitly crafted for the control or interaction with robotic arms, but it can be used in conjunction with various libraries and frameworks that are commonly employed in robotics, such as TensorFlow, PyTorch, or OpenCV. It provides free access to GPU and TPU resources, which can significantly accelerate the training of machine learning models. Colab offers a convenient environment for developing and testing algorithms related to the control and movement planning of a 6-DOF robotic arm.

Machine Learning: Through the machine learning process, robots gain the ability to recognize patterns, enabling them to comprehend their surroundings and execute tasks with greater efficiency based on learned knowledge. Utilizing machine learning algorithms, robots can autonomously acquire information without the need for bespoke programming for each individual task

Jupyter Notebook: Jupyter Notebook stands out as a widely embraced open-source tool for interactive computing. It empowers users to create and share documents seamlessly integrating live code, equations, visualizations, and narrative text.This is useful for developing and testing algorithms for controlling robotic arms, as it can iteratively make changes to the code and immediately see the results. Using Jupyter Notebooks in robotic arm research enhances the development process, facilitates documentation, promotes collaboration, and supports a more interactive and iterative approach to experimentation and analysis.

VS Code: Visual Studio (VS) is a comprehensive integrated development environment (IDE) expertly designed and developed by Microsoft,and it is widely used in various software development domains, including robotics.It supports multiple programming languages, including C++ and C#, which are commonly used in robotic arm programming. Visual Studio can be used for developing ROS nodes and packages, allowing developers to create and control robotic arms within the ROS ecosystem. Microsoft Robotics Developer Studio is an example of a tool that integrates with Visual Studio for simulation purposes.

In the software part, the most essential libraries used in the project are Moveit, rclpy, std_msgs, geometric_msgs and Adafruit_PWMServoDriver, SoftwareSerial, TinyGPSPlus, TensorFlow, OpenCV, NumPy, Matplotlib, Pytorch, Scikit-learn,

Moveit: MOVEit, a product by Ipswitch, Inc. (now part of Progress Software), stands as a sophisticated managed file transfer software. Offering encryption for files and leveraging file transfer protocols like FTP(S) or SFTP, MOVEit excels in data transfer. Additionally, it provides automation services, analytics, and failover options for a comprehensive file management experience.

Adafruit_PWMServoDriver: Adafruit_PWMServoDriver initializes a new PCA9685 PWM driver chip on a TwoWire interface with a specified I2C address.

SoftwareSerial: The SoftwareSerial library facilitates serial communication using a digital pin, providing an alternative to the default serial port. This library supports the creation of multiple software serial ports, each capable of operating at speeds up to 115200 bps. However, when employing multiple software serial ports, it's important to note that only one port can receive data at any given time.

TinyGPS: TinyGPS, created by Mikal Hart, translates NEMA format global positioning data from consumer GPS devices into easily accessible variables such as Latitude, Longitude, Time, Altitude, Speed, and Course. Communication between GPS modules and Arduino occurs through serial communication.

Std_msgs: std_msgs in ROS encompasses standard message types representing fundamental data types and basic message constructs, including multiarrays. ROS messages are characterized by both message type and data format. Additional ROS message packages cater to robot navigation and robotic sensor messages.

Geometry_msgs: geometry_msgs offers messages for prevalent geometric primitives like points, vectors, and poses. These primitives are structured to establish a standardized data type, fostering seamless interoperability across the system.

Tensorflow: TensorFlow stands out as a widely used framework for creating and deploying machine learning models. Conversely, ROS (Robot Operating System) serves as middleware, offering a unified platform for various robotics applications

OpenCV: OpenCV is a widely acclaimed open-source library extensively used in computer vision, machine learning, and image processing. Its prominence lies in its effectiveness in real-time operations, a critical component of contemporary systems. Employing OpenCV facilitates the manipulation of images and videos, enabling tasks like object detection, facial recognition, and even the analysis of human handwriting.

Numpy: NumPy, a fundamental package for scientific computing in Python, provides a versatile multidimensional array object and various derived objects. With a rich set of routines, NumPy facilitates rapid operations on arrays, including mathematical, logical, and shape manipulation functions, sorting, selection, input/output operations, discrete Fourier transforms, basic linear algebra, fundamental statistical functions, and random simulation, among other capabilities.

Matplotlib: Matplotlib, a cross-platform data visualization library, is constructed on NumPy arrays and designed to seamlessly integrate with the extensive SciPy stack. Known for its user-friendly interface, Matplotlib effortlessly replicates MATLAB-style graphs and visualizations.

PyTorch: PyTorch, an open-source machine learning (ML) framework, is built on the Python programming language and the Torch library.PyTorch is a fully featured framework for building deep learning models, which is a type of machine learning that's commonly used in applications like image recognition and language processing.

Hardware used for our project are:

- Raspberry PI 3 model B
- Arduino UNO
- ESP32
- 6 DOF Aluminum Mechanical Robotic Arm Clamp Claw Mount
- Servo Motor
- 16-channel servo driver
- Xbox Kinect one-sensor camera
- BTS7960 Motor Driver
- 12v, 300 RPM, 10kg Torque gear motor
- Wheel
- Motor Mount
- GPS Module
- GSM Module
- Bluetooth
- Step-down buck converter
- 12v 2200mAh Lipo Battery
- Lipo battery charger
- SS Sheet

Raspberry PI 3 Model B: This single-board device, resembling a low-cost credit card, possesses sufficient processing power to smoothly run applications like office suites (e.g., OpenOffice), graphic editing tools (e.g., GIMP), and various related programs. Its storage is facilitated through an SD card, eliminating the need for a built-in hard drive. The device is built around a Broadcom System on a Chip (SoC) featuring a GPU, an ARM processor with 26–52 MB of memory. Power is supplied through a 5 V USB cord, while display connectivity is supported through DVI/HDMI ports, with compatibility for HDMI cables or DVI converter HDMI cables. Input is facilitated by USB mouse/keyboard, and network connectivity is established through Ethernet.

Arduino UNO: The Arduino UNO is a microcontroller board featuring the ATmega328P microcontroller. It boasts 14 digital input/output pins, with 6 capable of functioning as PWM outputs, along with 6 analog inputs. Equipped with a 16 MHz ceramic resonator, the board also includes a USB connection, a power jack, an ICSP header, and a reset button. All essential components to support the microcontroller are integrated, allowing users to easily initiate it by connecting to a computer via a USB cable or powering it with an AC-to-DC adapter or battery. Programming the Arduino UNO is accomplished using the Arduino IDE (Integrated Development Environment) software.

ESP32: The ESP32 stands out as an economical and energy-efficient system on a chip series that incorporates integrated Wi-Fi and dual-mode Bluetooth capabilities. Depending on the module, it has a dual-core or single-core microcontroller with speeds ranging from 160 to 240 MHz. The ESP32 has up to 12 external interrupt pins that can be used for various purposes like wake-up, data transmission between chips, system events, etc. It also contains an embedded flash memory of 4MB to 16MB for Wi-Fi firmware and applications.

6 DOF Aluminum Mechanical Robotic Arm Clamp Claw Mount: This 6 DOF aluminum robotic arm provides a cost-effective solution for tasks requiring dexterous 3D movement. Its lightweight yet rigid construction comprises rotating joints that smoothly mimic real muscle motion. An adjustable claw clamp enables the grasping of various payloads with an integrated end effector. Standard servo connectors allow for simple programming and prototyping integration using common microcontrollers. Suitable applications include automation, robotics education, inspection, and delicate prototyping.

Servo Motor: The servo motor is a rotary actuator that provides precise angular control via position feedback. It consists of a motor coupled to an encoder and controller. Standard servos are compact yet produce adequate torque for rotational tasks like gripping or flipping objects. Programming position is done through PWM signals to the motor shaft. Applications include robotics, RC vehicles, automation, and prototyping due to their easy controllability and affordable pricing.

16-channel servo driver: This controller interfaces 16 servos with a microcontroller. It distributes PWM signals to each channel for individual/synchronized position control. Onboard filtering ensures stable motor operation. Suitable for advanced projects involving coordinated multi-motor movements like robots or machinery. Frees the microcontroller while enabling complex mechanisms with numerous rotational degrees of freedom.

Xbox Kinect One Sensor Camera: The Kinect provides an all-in-one solution for computer vision and motion tracking. Its high-definition RGB camera and infrared depth sensor enable full-body 3D scanning within 4m. Proprietary algorithms detect up to 6 skeletons with over 20 tracked joints in real time. Versatile SDK supports robotics, displays, scanning, sign language recognition, and more. An affordable price point makes the Kinect accessible for education, research, and creative applications beyond gaming.

BTS7960 Motor driver: The BTS7960 is a popular H-bridge motor driver IC capable of controlling two DC motors up to 1.5A continuously via PWM speed regulation and direction selection inputs. Short circuit and overload protection make it well-suited and safe for robotics, CNC, RC vehicles, and student projects operating from 4.5 to 18V. Thanks to its compact size and simple interfacing, the BTS7960 simplifies the creation of motorized mechanisms.

12v, 300 RPM, 10kg Torque gear motor: This compact 12V gear motor spins at 300 RPM while providing robust 10kg torque output through its reduction gears. Integrated limit switch protects from over-rotation. Well-suited for precision lifting, rotating, or positioning of loads up to 10kg in automated machinery, robotics, and more. Constant low-RPM operation prioritizes torque over speed. Screw mounts ease integration in tight spaces. An overall powerful yet small package for torque-focused actuation.

Wheel: The wheel is a fundamental circular component designed for efficient rolling motion. Consisting simply of a rim/ring connected via spokes/hub to a central axle, it excels at converting rotational force into usability. Wheels take many forms to suit diverse applications from transport to machinery.

Motor Mount: A motor mount is a mechanical component that securely attaches an electric motor while isolating vibrations. Common types include rubber, spring and inertia base mounts suited to motor size/weight. Mounts prevent structural vibrations from impacting the motor or transferring to other components. They secure the motor safely and allow wiring/maintenance access. Choosing the appropriate mount design based on vibration levels and other factors leads to smoother machinery operation. A critical yet overlooked detail for robust motor integration and precision mechanical function.

GPS Module: A GPS module is an electronic device that utilizes satellite signals to precisely calculate its location, providing latitude, longitude, and altitude data. Key components include a satellite signal receiver and an onboard processor. Common outputs integrate with displays, loggers, and external processors. Applications range from automotive and marine navigation to asset-tracking systems and drones. Feature selection considers sensitivity, accuracy, and interfaces. GPS modules enable the growing field of location-based technologies by providing accurate positioning data.

GSM Module: A GSM module is a wireless device that enables cellular connectivity through the Global System for Mobile Communications network. Key components include a SIM card interface, radio transceiver, and standard interfaces. Common applications are adding cellular data, calling, and SMS functions to devices through AT commands. Machine-to-machine and IoT systems also utilize GSM modules for remote monitoring and control via mobile networks. Frequency band and interface type selection ensure compatibility. GSM modules empower technologies with wireless communication capabilities.

Bluetooth: Bluetooth is a wireless standard utilizing the 2.4 GHz band to enable short-range data transmission between devices using UHF radio waves. Key specifications include protocols for radio, baseband, link management, and service discovery. Main applications include wireless audio in headphones/speakers, connectivity of peripherals like keyboards/mice, and data sharing between devices.

Step-down buck converter: The buck converter is a basic yet important DC-DC topology containing a switching MOSFET, inductor, and diode. It utilizes PWM to pulse the inductor voltage supplied from the DC source, inducing a changing magnetic field. The diode then rectifies this, outputting a regulated lower voltage. Buck converters can tightly control this output even with varying input/load conditions, making them well-suited for power supplies and battery-powered devices needing to step down voltages.

3s 12v 2200mAh Lipo Battery: The 3s 12v 2200mAh lithium polymer battery contains 3 cells in series, providing 11.1v that can surge to 12.6v. Its capacity allows sustained current draw before recharging is needed. Lightweight with high energy density, Lipos suits portable yet powerful applications, making the 3s configuration suitable for RC vehicles, drones, and more.

Lipo battery Charger: A Lipo charger is essential for safely charging lithium polymer battery packs. Most contain circuits controlling charging current, voltage cutoff points and timing to match the battery. Features include adjustable rates from 0.1A to 1C or more, plus balanced charging for multi-cell packs.

SS Sheet: Stainless steel sheet contains predominantly iron with chromium, usually 18%, added for corrosion resistance via a passive oxide layer. Sheets are available in grades like 300, 400, and 200 series with differing rust protection suitable for various applications. Sheets can be acquired in an array of thicknesses and surface finishes, making fabrication of appliances, food processing equipment, and more possible while maintaining integrity for years.

Screw: The screw is a simple yet essential fastener. It consists of a threaded steel shaft with a drive recess-like slot, Phillips, or hex in the head. Common thread gauges include #6-32, #8-32, and #10-24 (diameter, threads per inch). Machine screws securely join parts by cutting threads into mating holes during tightening.

3.2 Robot Design

3.2.1 6 DOF Robotics Arm

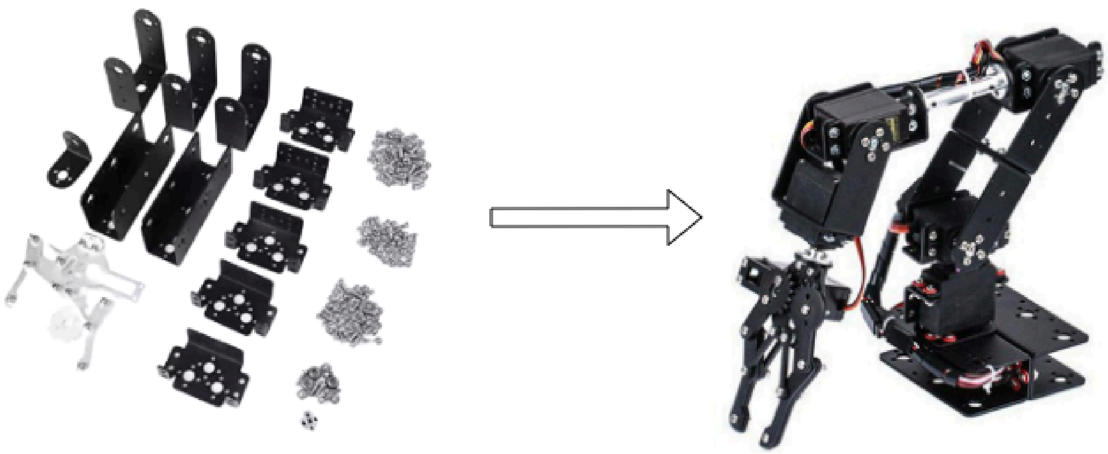


Fig 3.1: Robotics Arm Design Using Aluminum Clamp Claw mount

EnviroBot features a 6 Degrees of Freedom (6 DOF) robotic arm, carefully chosen for its versatility and cost-effectiveness, especially in the context of Bangladesh. The 6 DOF robotic arm is a key element in the design, providing the necessary range of motion for efficient waste collection and disposal. Here is an in-depth look at the design considerations:

1. **6 DOF Structure:**
- The robotic arm comprises six joints, each representing a degree of freedom. This design allows for intricate and precise movements, crucial for navigating diverse environments and collecting waste with accuracy.

2. **Servo Motors Integration:**
- Each joint of the robotic arm is powered by servo motors, which provide controlled and precise movements. The integration of servo motors allows for the manipulation of the arm in multiple directions.

3. **Aluminium Claw and Clamp:**
- To facilitate the effective gripping and collection of waste, an Aluminium Claw and Clamp mechanism is incorporated at the end of the robotic arm. This design choice ensures a secure grip on various types of waste materials.

4. **Versatility and Cost-Efficiency:**
- The selection of a 6 DOF robotic arm aligns with the project's goals of versatility and cost-efficiency. This type of arm strikes a balance between functionality and affordability, making it suitable for the intended application in waste management.

5. **Connectivity:**
- The joints of the robotic arm are interconnected to form a seamless structure. The connectivity allows for coordinated movements, enabling the arm to reach and collect waste efficiently.

The design of the 6 DOF robotic arm prioritizes functionality, cost-effectiveness, and adaptability to the waste collection tasks envisaged for EnviroBot. The incorporation of servo motors and an Aluminium Claw and Clamp ensures that the robotic arm can perform the required actions with precision, contributing to the overall effectiveness of the waste management system.

3.2.2 4 Wheel Rover

Fig 3.2: SS Sheet, Gear Motor and Wheel

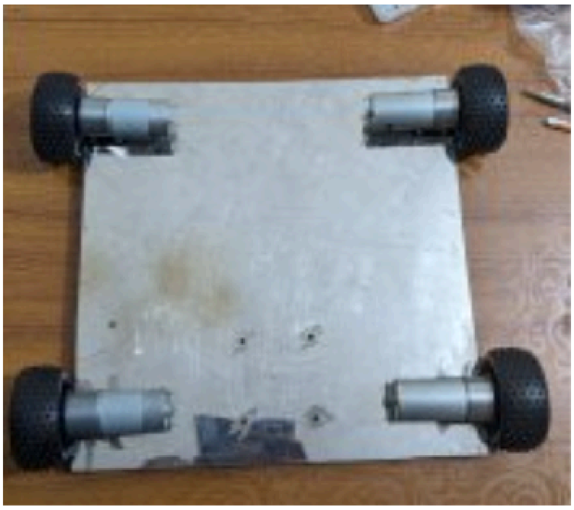



Fig 3.3: Motor attached to the SS sheet

In order to address the weight and mobility challenges posed by the heavy robotic arm, EnviroBot incorporates a purpose-built 4-wheel rover. This rover serves as the mobile platform for the robotic arm, ensuring its efficient transportation to different locations. Here are the key design considerations for the 4-wheel rover:

1. **Heavy-Duty Design:**

- The robotic arm's weight necessitated the creation of a sturdy and robust rover. To meet this requirement, a 14 by 14 inches base was constructed using stainless steel sheet material. This material choice ensures durability and strength, allowing the rover to carry the heavy load of the robotic arm.

2. **Motorized Mobility:**

- Four gear motors, each with a torque of 10 kg and a speed of 300 rpm, were strategically chosen for the rover's propulsion. This motor configuration provides the necessary power to navigate various terrains, including rough and uneven surfaces.

3. **Secure Motor Installation:**

- The motors are securely installed on the stainless-steel base, ensuring stability and minimizing the risk of damage during movement. This secure installation is crucial for maintaining the structural integrity of the rover.

4. **Versatile Movement:**

- The 4-wheel design offers versatility in movement, allowing the robot to navigate through diverse environments. This ensures that the combined system, consisting of the robotic arm and the rover, can effectively reach and operate in different locations

5. **Integration with Robotic Arm**:
- The rover serves as the basement for the robotic arm, providing the necessary mobility for waste collection in various settings. The integration ensures seamless coordination between the rover and the robotic arm.

By integrating a dedicated 4-wheel rover into the design, EnviroBot gains enhanced mobility, allowing it to traverse rough terrains with the heavy robotic arm in tow. This design choice contributes to the adaptability and effectiveness of the waste management system in diverse environments.

3.2.3 Integration of Robotic Arm and Rover:

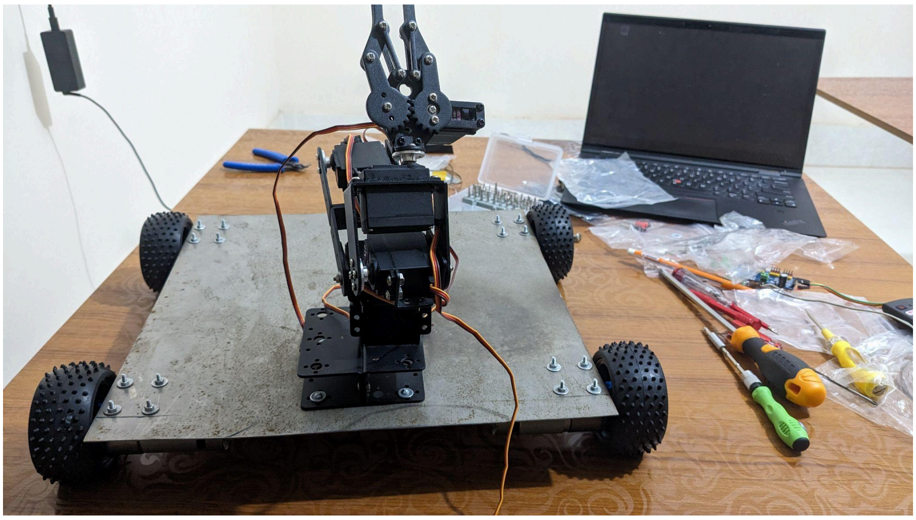


Fig 3.4: Combine Robotics Arm with Rover basement.

Ensuring effective waste collection requires strategic placement and integration of the robotic arm with the rover. EnviroBot has been meticulously designed to optimize the placement of the robotic arm on the 4-wheel rover. Here are the key considerations for the integration:

1. **Strategic Placement:**
- The robotic arm is strategically positioned on the stainless-steel sheet, ensuring an optimal placement that facilitates efficient waste collection. The placement is carefully chosen to enable the arm to reach and collect waste effectively.

2. **Front-Facing Placement:**
- To enhance the arm's reach and collection capabilities, it is fixed to the front of the 4-wheel rover basement. This front-facing configuration allows the arm to approach waste directly, ensuring proper and thorough collection.

3. **Unrestricted Movement:**
- Placing the robotic arm at the front of the rover eliminates obstacles that could hinder its movement. This design choice ensures that the arm has unobstructed access to waste, enabling it to perform its collection tasks without limitations.

4. **Enhanced Maneuverability:**
- By fixing the arm at the front of the rover, EnviroBot gains enhanced maneuverability. The front placement allows the robot to approach waste from multiple angles, adapting to the specific requirements of different collection scenarios.

This intentional integration of the robotic arm at the front of the 4-wheel rover showcases a thoughtful design approach, prioritizing the effectiveness of waste collection. The placement on the stainless-steel sheet ensures stability and durability, creating a cohesive system that excels in navigating diverse environments for optimal waste management.

3.3 Testing and Calibration

3.3.1 Motor testing

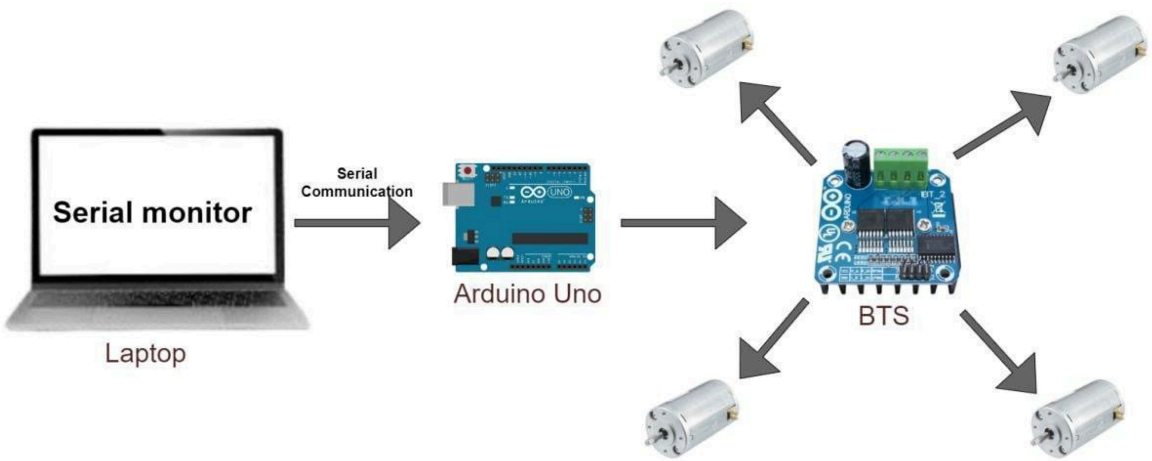


Fig 3.5: Motor Testing Data Flow Diagram

Before proceeding with the integration of various components, a crucial step involves testing the motors and motor drivers to ensure proper functionality. This step-by-step process ensures that each motor operates in the correct direction and is seamlessly integrated with the motor driver.

1. **Motor Driver and Motor Connection:**

- Connect the motor driver to the Arduino board, ensuring correct wiring and connections. Refer to the motor driver and motor specifications to guarantee proper interfacing.

2. **Arduino Serial Monitor Testing:**

- Utilize the Arduino Serial Monitor to monitor the output related to the motor driver. This step is essential for identifying any issues with the motor or its connection.

3. **Direction Verification:**

Examine the motor driver output in the Serial Monitor to verify the direction of the connected motor. Ensure that the motor rotates in the expected direction based on the control signals.

4. **Corrective Actions:**

- If any discrepancies are observed, such as inverted motor direction, take corrective actions by adjusting the motor connections or modifying control signals in the Arduino code.

By systematically testing the motors and their drivers, potential issues can be identified and resolved early in the development process, ensuring the smooth operation of the entire robotic system. This step lays the foundation for subsequent integration and testing phases.

3.3.2 Servo Calibration

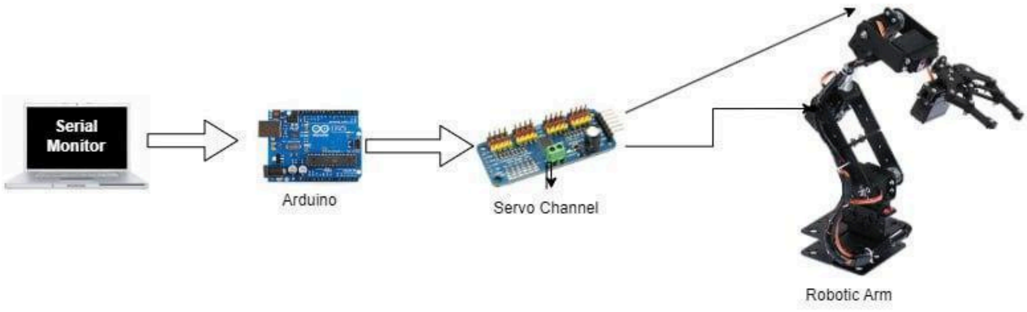


Fig 3.6: Servo Calibration Data Flow Diagram

Servo calibration is a critical step to ensure that each joint of the robotic arm operates within the desired range of motion. This process involves measuring the initial, maximum, and minimum turning angles for each servo motor. The Arduino Serial Monitor is utilized to facilitate this calibration:

1. **Calibrate Initial Position:**
- Gradually move each servo motor to its initial position and record the corresponding angle values displayed on the Serial Monitor. This establishes the baseline for each joint.

2. **Determine Maximum and Minimum Turns:**
- Move each servo motor to its maximum and minimum turning angles while noting the angle values in the Serial Monitor. This step defines the range of motion for each joint.

3. **Validate Consistency:**
- Confirm that the servo motors exhibit consistent and predictable behavior across multiple calibration cycles. Inconsistencies may indicate the need for adjustments or troubleshooting.

4. **Adjustments if Necessary:**
- If any servo motor deviates from the expected range or exhibits erratic behavior, make necessary adjustments to the physical connections or calibration code.

5. **Documentation:**
- Document the calibrated values for each servo motor, specifying the initial, maximum, and minimum turning angles. This documentation serves as a reference for future control and programming tasks.

By systematically calibrating the servo motors using the Arduino Serial Monitor, precision and accuracy in the robotic arm's movements are ensured. This calibration data becomes crucial for programming and controlling the robotic arm during waste collection tasks.

3.4 Control using Mobile apps and Bluetooth

Enabling manual control of the robotic arm and rover through a mobile app using Bluetooth is a key step towards achieving eventual autonomy. This section outlines the steps involved in implementing the Bluetooth control system:

3.4.1 MIT Apps Inventor

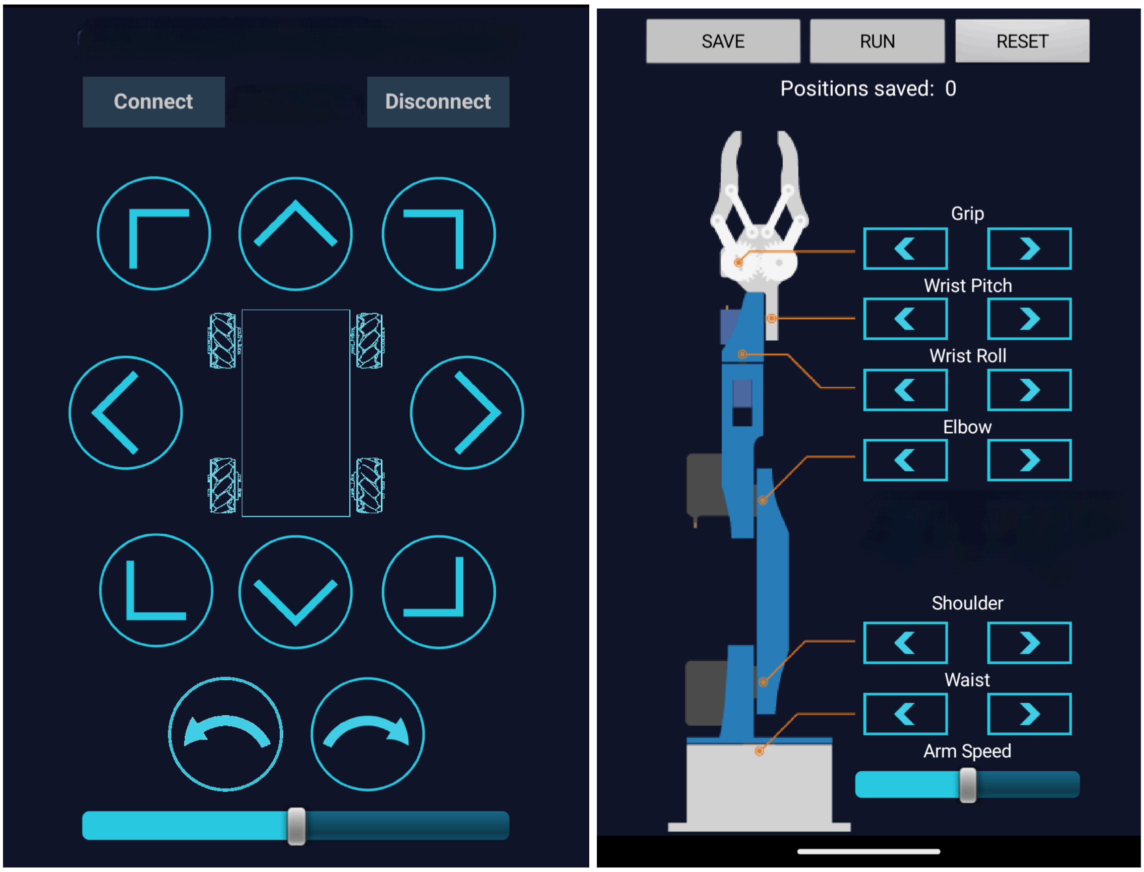


Fig 3.7: MIT app for control our robot

MIT App Inventor, a visual programming platform, to expedite the development of a mobile application for wireless control of the EnviroBot robotic arm. The integration aims to provide a user-friendly interface, enabling immediate control via Bluetooth communication.

1. **Designing the User Interface:**

Design an intuitive user interface within MIT App Inventor. The interface incorporates buttons and sliders, each corresponding to specific movements and actions of the robotic arm. The visual designer allows for easy arrangement and customization of controls, enhancing the user experience.

2. **Bluetooth Component Integration:**

MIT App Inventor facilitates the integration of the Bluetooth component, a critical element for wireless communication with the EnviroBot robotic arm. The platform's blocks-based coding paradigm simplifies the setup of Bluetooth pairing, connection establishment, and data transmission.

3.4.2 Bluetooth Control

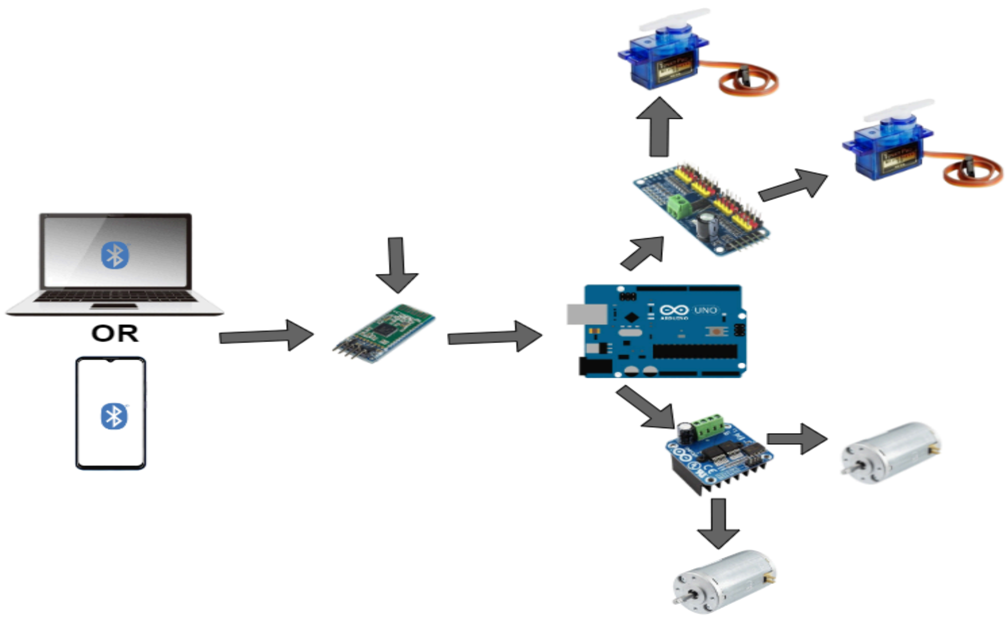


Fig 3.8: Bluetooth Control Approach Data Flow Diagram

These steps were taken to enable manual control of the EnviroBot robot through a mobile application developed using MIT App Inventor. The integration involves sending control commands from the app to the Arduino board, which, in turn, interprets these commands and actuates the movements of both the wheels and the robotic arm

1. **Bluetooth Communication Setup:**

   A Bluetooth module is integrated with the Arduino board to establish wireless communication between the mobile app and the robotic system. The Bluetooth module is configured to receive data transmitted from the mobile app.

2. **App Commands and Data Transmission:**

   The mobile app is programmed to generate specific commands based on user interactions. These commands are translated into data packets and transmitted via Bluetooth to the connected Arduino board.

3. **Iterative Testing and Optimization:**

   The integration undergoes rigorous testing to ensure the accuracy and responsiveness of the manual control system. Iterative optimizations are made to both the mobile app and the Arduino code to enhance user experience and system performance.

The manual control system lays the foundation for future developments, such as autonomy and task-specific functionalities

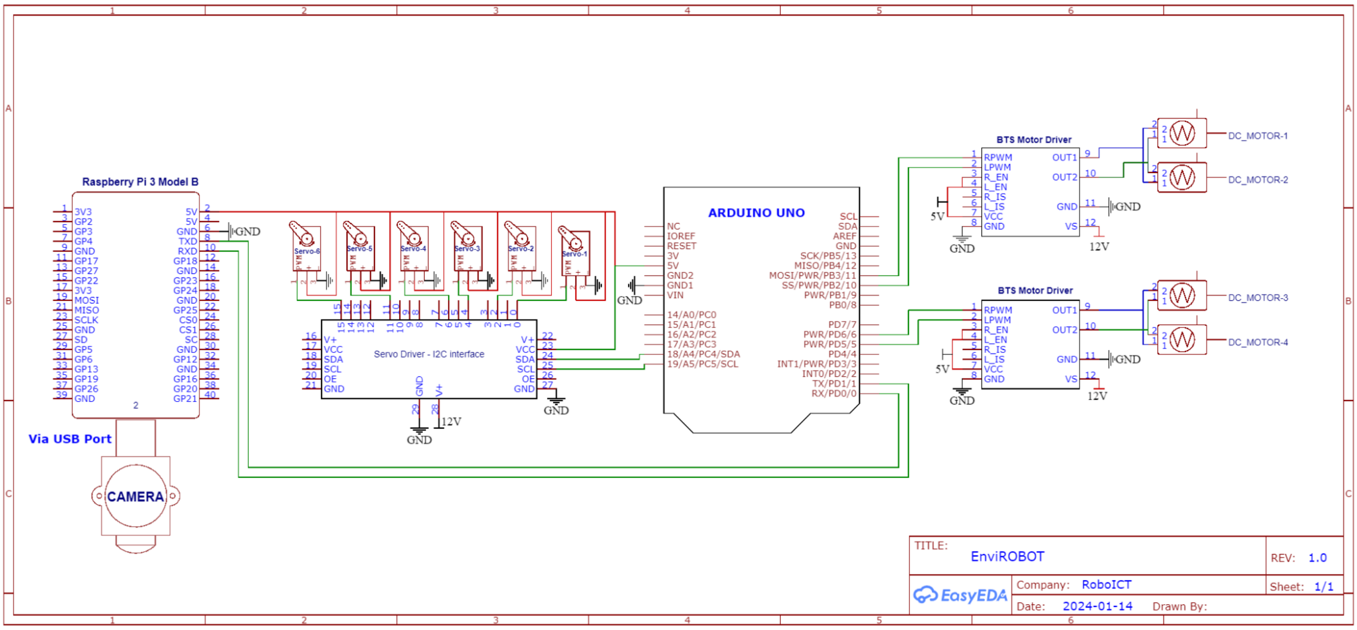


Fig 3.9 : Circuit Diagram

3.5 Autonomous Waste Collection

3.5.1 ROS2 Environment Setup

ROS2 Installation:

- Installed ROS2 by following the official installation instructions for the chosen distribution

Workspace Initialization:

- Created a new ROS2 workspace using the `colcon` build system.
- Executed the following commands to set up the workspace:

```
mkdir -p ~/ros2_workspace/
cd ~/ros2_workspace
Colcon build
source install/setup.bash
```

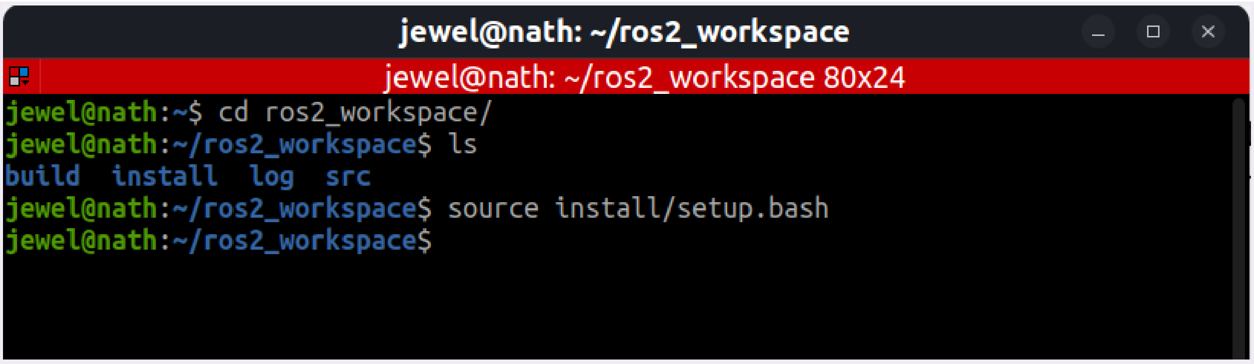


Fig 3.10: Ros2 Workspace creating

Dependency Installation:

- Installed additional ROS2 dependencies specific to the project requirements, adhering to the guidance provided in the ROS2 documentation.

3.5.2 Development Environment Configuration

Environment Variables:

- Configured environment variables to ensure proper ROS2 package sourcing by adding the following lines to the shell configuration file

```
source /opt/ros/humble/setup.
```

```
source ~/ros2_ws/install/setup.bash
```

- Sourced the updated configuration using `source ~/.bashrc` to apply changes.

Integration Testing:

- Utilized the `ros2 doctor` command to perform initial integration testing of the ROS2 environment.
- Addressed any reported issues to guarantee a robust ROS2 setup.

3.5.3 Version Control and Collaboration

Version Control System:

- Integrated a version control system (e.g., Git) to manage the project's source code and ROS2 packages.

Collaborative Development:

- Collaborated with team members by hosting the ROS2 project repository on a version control platform

3.5.4 Robot visualization in Rviz

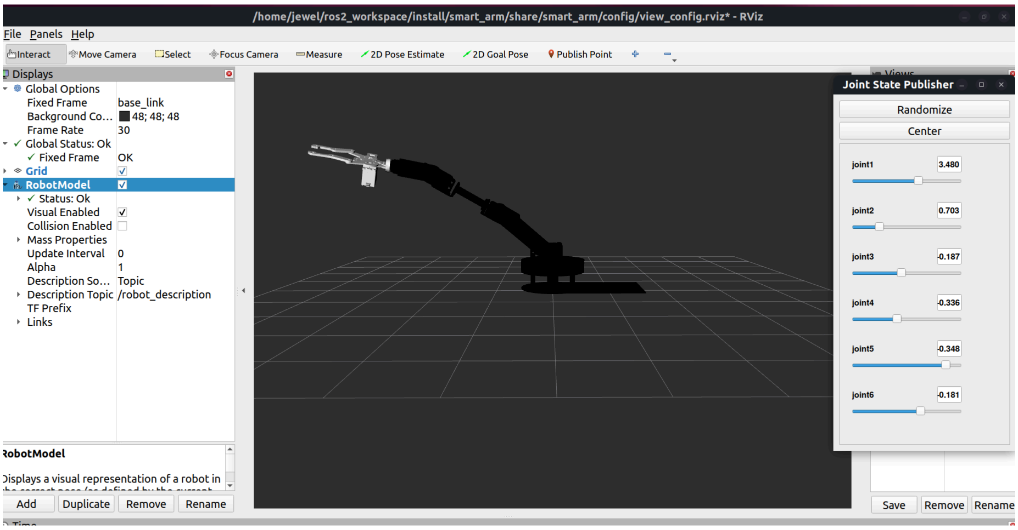


Fig 3.11: Robot URDF Visualize in Rviz

Robot visualization is a crucial aspect of our project, providing a comprehensive understanding of the robot's state, sensor data, and planning information. The Robot Visualization (Rviz) tool, integrated with the ROS2 framework, plays a central role in this aspect.

3.5.5 Rviz Configuration and Setup

Package Integration:

- Incorporated the necessary ROS2 packages for robot visualization by including the appropriate dependencies in the project's `package.xml` and `CMakeLists.txt` files.

Launch Files

- Developed launch files to configure Rviz for the specific robot and sensor setup.
- Configured parameters such as robot model, sensor displays, and coordinate frames to ensure accurate visualization.

3.5.6 Components of Visualization

Robot Model:

- Imported the robot model into Rviz using the URDF (Unified Robot Description Format).
- Verified the accuracy of the robot model's representation in Rviz to align with the physical robot.

Sensor Data Visualization:

- Integrated sensor data (e.g., camera images, point clouds) into Rviz for real-time visualization.
- Configured appropriate Rviz displays to visualize sensor outputs, aiding in debugging and analysis.

Planning Information:

- Displayed planning information, such as trajectories and waypoints, in Rviz to validate and visualize the robot's planned movements.
- Utilized Rviz's interactive markers and tools to modify and refine planning parameters.

3.5.7 Dynamic Visualization during Execution

Real-time State Updates:

- Enabled real-time updates of the robot's state in Rviz during execution, ensuring a dynamic and accurate representation.

Interactive Marker Usage:

- Leveraged Rviz's interactive markers for user interaction during development and testing, allowing for intuitive adjustments to the robot's pose and planning parameters.

3.5.8 Collaboration and Debugging

Collaborative Visualization:

- Shared Rviz configurations and visualizations with team members to facilitate collaborative development and debugging.

Debugging Tools:

- Utilized Rviz's built-in tools for debugging, such as visualizing TF (Transform) frames and inspecting topics, to identify and resolve issues.

3.5.9 MoveIt Configuration for Robotic Arm Motion Planning

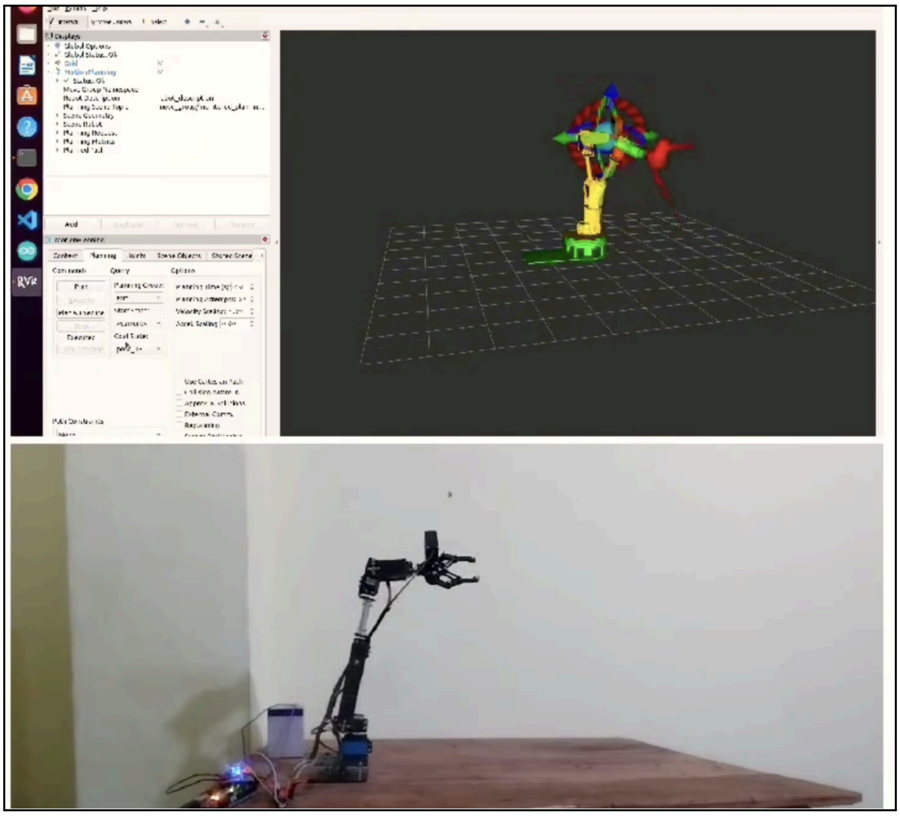


Fig 3.12: Motion Planning using moveIt

MoveIt, a powerful motion planning framework integrated with ROS2, has been instrumental in configuring motion planning for our robotic arm project. This section provides an overview of the MoveIt configuration, including how it was set up for trajectory generation, collision checking, and execution.

3.5.10 MoveIt Configuration Setup

Installation:

- Installed the MoveIt package for ROS2 using the appropriate ROS2 distribution.
- Ensured that MoveIt dependencies and additional packages were installed to support the specific robotic arm.

URDF Integration:

- Incorporated the robot's URDF (Unified Robot Description Format) into the MoveIt configuration to establish the robot's kinematic and dynamic properties.

MoveIt Setup Assistant:

- Utilized the MoveIt Setup Assistant to streamline the configuration process.
- Configured the robot's planning groups, end effectors, and defined the virtual representation of the robot in MoveIt.

3.5.11 Motion Planning and Trajectory Generation

Joint and Cartesian Planners:

- Configured both joint-space and Cartesian-space planners to provide flexibility in motion planning.
- Adjusted planner parameters to suit the robot's kinematics and dynamics.

Trajectory Generation:

- Defined and configured the trajectory generation settings, considering joint limits, velocity constraints, and acceleration constraints.
- Tested and tuned trajectory generation parameters to ensure smooth and collision-free movements.

3.5.12 Collision Checking and Avoidance

Collision Matrix:

- Established a collision matrix to specify which robot components should be considered in collision checking.
- Defined allowed and forbidden collision pairs to ensure safety during motion planning.

Self-Collision and Environment Collision Checking:

- Enabled self-collision checking to prevent the robot from colliding with its own components.
- Configured collision checking against the environment, including obstacles and other objects in the workspace.

3.5.13 Execution and Monitoring

MoveIt Controller Configuration:

- Integrated and configured controllers for executing trajectories generated by MoveIt.
- Adjusted controller parameters to achieve accurate and responsive execution.

Real-time Monitoring:

- Implemented real-time monitoring of the robot's state during trajectory execution using MoveIt's built-in visualization tools.
- Used Rviz to visualize planned trajectories and executed motions for debugging and analysis.

3.5.14 Autonomous Pick and Place Implementation

Implementing autonomous pick and place capabilities in our robotic arm project involves integrating perception, planning, and control modules. This section provides an overview of the key components and processes involved in achieving autonomous manipulation tasks.

1. **Perception Module**

Sensor Integration:

- Integrated camera and depth sensor to capture information about the environment.
- Configured sensor data processing pipelines to extract meaningful features for object recognition.

Object Recognition:

- Implemented computer vision algorithms to recognize and classify
- objects within the robot's workspace.
- Fine-tuned object recognition models to improve accuracy and robustness.

2. **Planning Module**

Path Planning:

- Implemented path planning algorithms to generate collision-free trajectories for the robot's end effector.
- Utilized MoveIt for motion planning, considering joint limits, obstacle avoidance, and dynamic constraints.

Grasping Strategy:

- Developed a grasping strategy to determine the optimal approach and pose for picking up objects.

- Considered object properties (shape, size, orientation) in the planning process.

3. **Control Module**

Trajectory Execution:

- Integrated the planned trajectories with the robot's control system to execute pick and place actions.
- Configured controllers for precise and smooth motion during execution.

Closed-Loop Control:

- Implemented closed-loop control to monitor and adjust the robot's position and orientation during the pick and place operation.
- Utilized feedback from sensors to enhance the accuracy of the manipulation tasks.

4. **Autonomous Task Execution**

Pick and Place Sequencing:

- Orchestrated the pick and place tasks into a coherent sequence for autonomous operation.
- Implemented logic to handle different scenarios, such as failed grasps or unexpected obstacles.

Error Handling:

- Implemented robust error handling mechanisms to recover from failures, such as unsuccessful grasps or collisions.
- Integrated feedback mechanisms to notify the system of task completion or issues.

5. **Real-world Testing and Validation**

Simulation Testing:

- Conducted initial testing in simulation environments to validate the pick and place algorithms and logic.
- Adjusted parameters and algorithms based on simulation results.

Physical Robot Testing:

- Tested the autonomous pick and place system on the physical robotic arm.

- Iteratively refined the system based on real-world performance and challenges

3.5.15 Waste Image Collection

1. **Dataset Description**

The dataset used for training the waste detection model in our project, named EnviroBot, comprises 1700 images categorized into three main types: Plastic, Paper, and Trash. This dataset was collected from a diverse range of sources, with a significant portion sourced from a previous senior batch project within our department.

2. **Collection Process**

Image Categories:

- The dataset was carefully curated to represent the diverse range of waste items encountered in real-world scenarios. The three primary categories, Plastic, Paper, and Trash, were chosen to cover common waste materials.

Source Diversity:

- The dataset includes images sourced from various environments, such as indoor spaces, outdoor landscapes, and public areas. This diversity ensures that the waste detection model is robust across different settings.

3. **Data Preprocessing**

Image Annotation:

- Each image in the dataset was meticulously annotated to label the waste items present. Annotations include bounding box coordinates and class labels for Plastic, Paper, and Trash.

Quality Assurance:

- Quality assurance measures were implemented during the annotation process to ensure accuracy. Annotators underwent training to adhere to consistent labeling standards.

4. **Image Augmentation**

Image augmentation is a process by which we can create new images from existing images. For increasing the number of images we adjusted the brightness of some images, shifted some 16 images horizontally and vertically, and rotated a few images. Image augmentation allows us to increase the size of our training set, by providing more data to our model for training

5. **Image Resizing**

Resizing images means changing the dimensions of images such as changing width or height. We have resized images in 512 x 384 for training the models. We have used the OpenCV library of Python for resizing the images. We provided 512 for width and 384 for height in pixels for resizing the pictures.

6. **Dataset Utilization**

Training Set:

- The dataset was divided into training, validation, and testing sets. The training set was used to train the YOLO waste detection model, while the validation and testing sets were employed for evaluation.

Model Evaluation:

- The performance of the trained model was rigorously evaluated using metrics such as precision, recall, and F1 score on the testing set to ensure robust waste detection capabilities

3.5.16 Yolo Model

YOLO v8: YOLO v8 was released in January 2023 by Ultralytics. YOLOv8 supports multiple vision tasks such as pose estimation, object detection, tracking, classification, and segmentation. YOLOv8 uses binary cross-entropy for classification loss and DFL loss functions and CIoU for bounding box loss. These losses have improved object detection performance, especially dealing with smaller objects. [24]

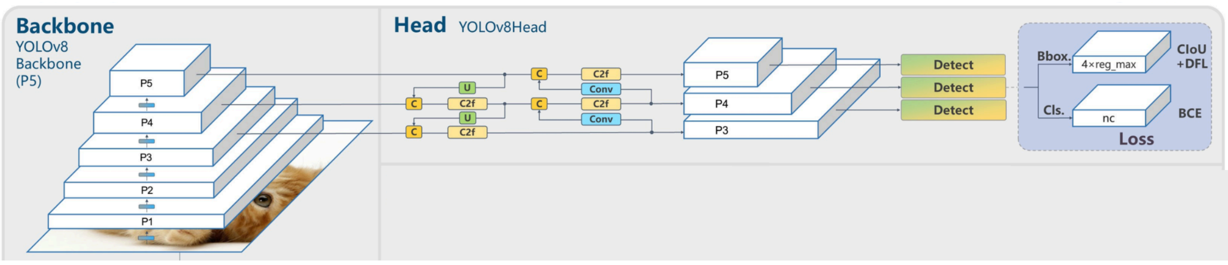


Fig 3.13 : Yolov8 architecture[25]

YOLO Model Training for Waste Detection:

The YOLO (You Only Look Once) model is employed for waste detection in the collected image dataset. The training process involves dividing the dataset into training, validation, and testing sets, with specific proportions allocated for each. The YOLO v8 model is trained over a specified number of epochs.

1. **Dataset Splitting:**

   The collected image dataset is divided into three sets:

   - Training Set (70%): 1226 images are used to train the YOLO model.
   - Validation Set (20%): 350 images are reserved for fine-tuning model hyperparameters.
   - Test Set (10%): 176 images are used for evaluating the model's performance on unseen data.

2. **YOLO Model Selection:**

   The YOLO v8 model is selected for its efficiency in real-time object detection. Its ability to detect multiple objects in a single pass makes it suitable for the dynamic environment of waste detection. We used Yolov8l (large model) for the training of our dataset.

3. **Model Training:**

   The YOLO model is trained on the annotated dataset, with the training set used to update the model's parameters. The validation set aids in fine-tuning, preventing overfitting and ensuring the model generalizes well to unseen data.

3.5.17 Embedded Detection Model with Robotics Arm

In this stage, the machine learning model for waste detection, specifically the YOLO model, is embedded into the EnviroBot robotic system. The integration involves connecting the detection model with the ROS2 MoveIt framework for motion planning. This allows the robotic arm to autonomously respond to waste detection, plan motions, and carry out pick-and-place operations.

ROS2 Node for Detection Model:

Develop a ROS2 node that acts as an interface between the YOLO detection model and the rest of the robotic system. This node is responsible for receiving input from sensors, invoking the detection model, and providing detected object information to other nodes.

1. **Real-time Data Transmission:**

- Establish a real-time data transmission pipeline from the camera system to the ROS2 nodes, ensuring minimal latency in waste detection.

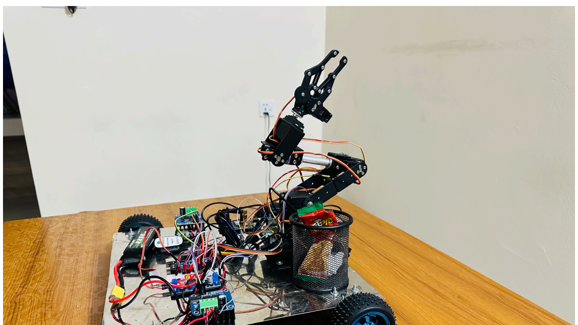
2. **Object Detection Coordinates:**
    - Extract the coordinates and classification information of detected waste objects from the output of the YOLO model. These coordinates will be crucial for motion planning.

3. **MoveIt Configuration:**
    - Configure MoveIt to accommodate the new sensory input from the YOLO detection model. Adjust the MoveIt configuration to account for the additional layer of intelligence in waste detection.

4. **Joint Trajectory Planning:**
    - Develop a joint trajectory planning mechanism that takes the waste detection coordinates as input. This planning module generates trajectories for the robotic arm to navigate to the detected waste locations.

Chapter - 4

RESULT & FINDINGS



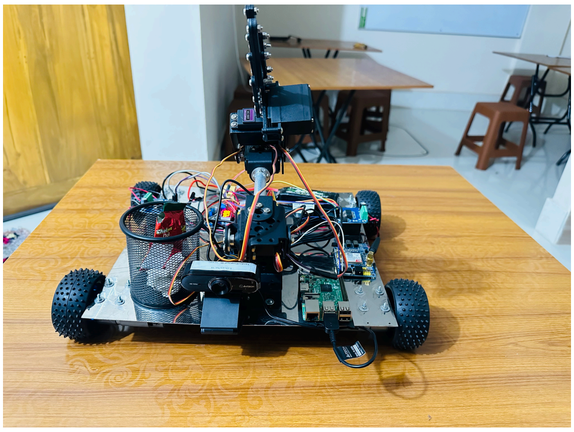


Fig 4.1: EnviroBot

4.1. Enhanced Mobility:

The implementation of enhanced mobility features in our EnviroBot project has yielded remarkable results. The four-wheel robotic platform now exhibits precision in movement, showcasing the ability to navigate both smooth and rough terrains with ease. Moreover, the platform's increased load-bearing capacity allows it to efficiently carry substantial amounts of waste, enhancing its operational efficiency in waste collection scenarios.

4.2. Flexible Robotics Arm:

our groundbreaking Flexible 6 Degrees of Freedom (6-DOF) Robotic Arm – an innovation in waste management! This advanced robotic arm, designed specifically for trash collection, showcases six articulated joints, delivering unmatched precision and adaptability in navigating varied environments.

Customized to meet the challenges of waste collection, our cutting-edge robotic arm seamlessly navigates urban environments, effortlessly accessing confined spaces and ensuring a meticulous cleanup. Whether it's picking up litter in public spaces, managing waste in challenging terrains, or optimizing recycling processes, our robotic arm is engineered to excel.

Ideal for municipalities, waste management facilities, and environmentally-conscious organizations, this trash-collecting robotic arm offers a sustainable solution with its precision engineering and user-friendly controls.

4.3 Performance of YOLO v8

The integration of the YOLO v8 model for waste detection has demonstrated commendable results. With an accuracy rate of 80%, the model showcases robust performance in identifying and classifying waste items. The dataset, carefully split into training, validation, and testing sets, enabled the YOLO v8 model to converge effectively over 50 epochs, with accuracy consistently improving.

Training Progress: The model's accuracy steadily increased, while the loss consistently decreased over the epochs, particularly after the 30th epoch. This indicates the model's ability to continuously learn and adapt to the intricacies of waste detection.

There is fluctuation in values of loss and accuracy. The accuracy and loss is shown in figure and 4.2 figure 4.3 respectively.

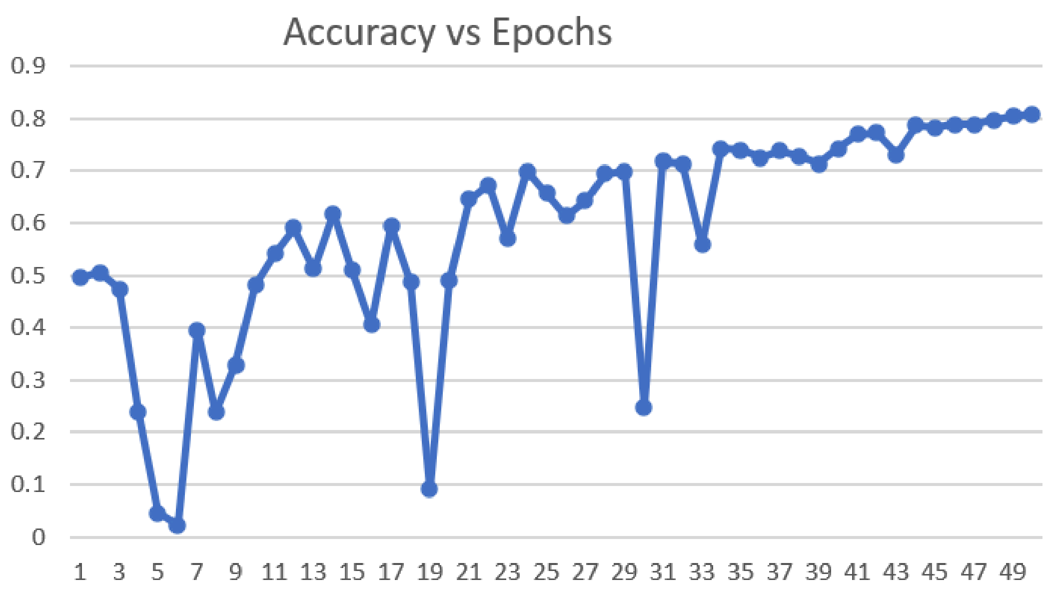


Fig 4.2: Accuracy vs Epochs

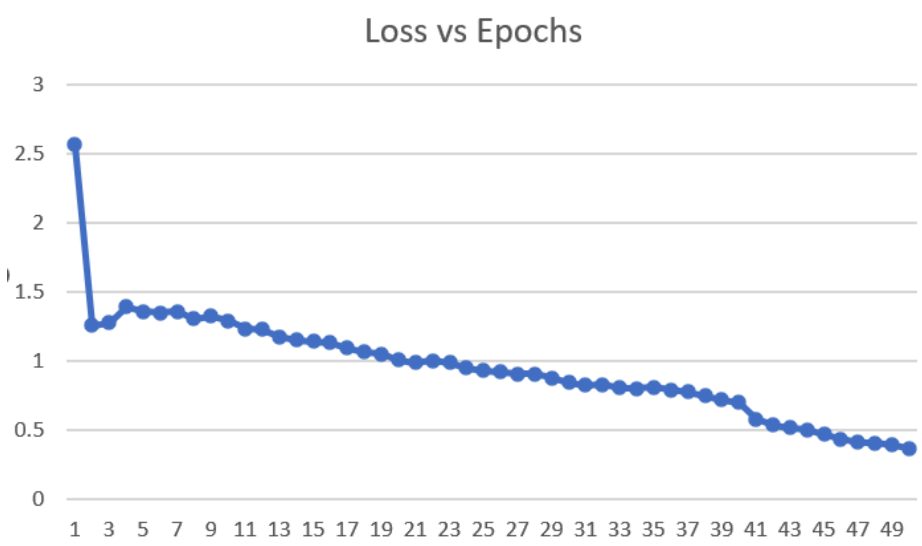


Fig 4.3: Loss vs Epochs

The correct classification and total misclassification are shown in a confusion matrix in the figure 4.4 From this figure we can say, mostly misclassified objects are 'trash'. Mostly classified objects are 'plastic'.

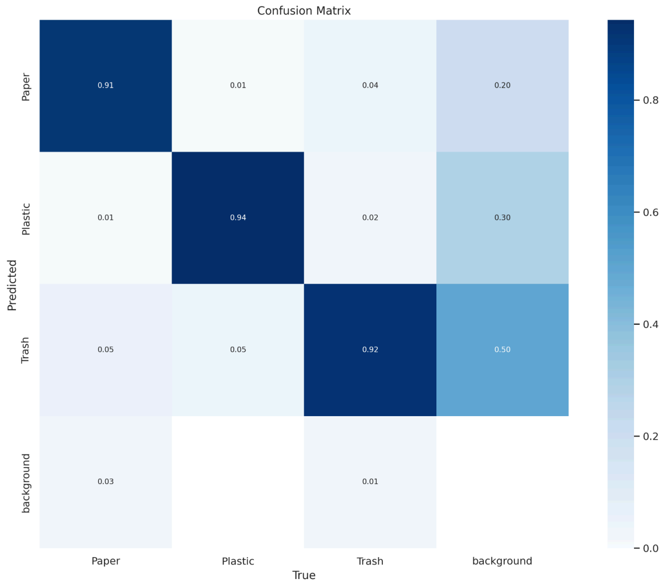


Fig 4.4: Confusion matrix for YOLO v8

The accurate and inaccurate predictions done by the YOLO v8 model are shown in figure 4.10 and figure 4.11, respectively.

Fig 4.5: Accurate prediction by YOLO v8

Fig 4.6: Inaccurate prediction by YOLO v8

4.4 Efficient Cleaning

EnviroBot's operational efficiency is evident in its ability to clean the environment with high efficiency. The integration of enhanced mobility, a flexible robotic arm, and a powerful waste detection model collectively contributes to the system's efficiency in waste collection and cleanup processes. In conclusion, the EnviroBot project's results highlight advancements in mobility, the innovative design of the robotic arm, the robust performance of the YOLO v8 model, and the overall efficiency in environmental cleaning operations. These findings position EnviroBot as a cutting-edge solution for sustainable and effective waste management.

Chapter - 5

Conclusion and Future Work

5.1 Conclusion

Our project aimed at establishing a dependable system for the efficient gathering and disposal of trash, utilizing a four-wheeled robotic arm designed for mobility and purposeful trash collection. The implementation of a streamlined object identification algorithm, leveraging camera technology, has been pivotal to our success.

Key Achievements:

Efficient Waste Collection: The integration of a robotic arm has enabled the systematic collection of diverse types of trash, including plastic, paper, and general waste. The ability to deposit these items into separate bins reflects a well-organized and efficient waste disposal system.

Environmental Impact: The significance of enhancing waste disposal practices is underscored by the potential benefits for environmental preservation. The prevalent issue of indiscriminate trash disposal poses a severe threat to the ecosystem, with non-biodegradable materials contaminating soil and water. Our responsible approach to collecting and handling discarded trash contributes to a cleaner environment and addresses this critical environmental challenge.

In conclusion, our project stands as a testament to the transformative power of technology in waste management. By combining the capabilities of a robotic arm with an efficient object identification algorithm, we have taken a significant step towards mitigating the adverse effects of indiscriminate trash disposal. The vision of a cleaner, more sustainable environment is within reach, and our project serves as a positive contribution to this endeavor.

5.2 Future Work

1. **Lidar Integration:**

   Future developments can incorporate Lidar technology, enabling the robot to map and navigate its working area with enhanced precision. Lidar's capabilities in generating detailed 3D maps can contribute to improved spatial awareness, facilitating more efficient and adaptive navigation.

2. **Depth Camera Enhancement:**

The integration of depth cameras holds promise in refining waste detection within the machine learning model. Depth cameras can aid in accurately detecting waste and determining their exact locations. This enhancement can further optimize the waste collection process, ensuring a more thorough and precise cleanup.

3. **Industrial Robotics Arm Integration:**

   Exploring the integration of high-quality industrial robotics arms presents an opportunity to enhance flexibility in operations. These advanced arms bring improved durability, precision, and adaptability to the system, offering a higher degree of flexibility in handling diverse tasks related to waste management. Integrating such arms can elevate the overall performance and capabilities of the robotic system.

These future work considerations aim to advance the capabilities of the robotic system, incorporating cutting-edge technologies for improved navigation, waste detection, and overall operational flexibility. The integration of Lidar, depth cameras, and industrial-grade robotics arms can contribute to a more sophisticated and efficient waste management solution.

REFERENCES

[1]Chang, Q., Liu, X., Xu, W., Yan, L. and Yang, B., 2016, October. The design and experiments of a small wheel-legged mobile robot system with two robotic arms. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 2590-2595). IEEE.

[2] Nantsios, G., Baras, N. and Dasygenis, M., 2021. Design and implementation of a robotic arm assistant with voice interaction using machine vision. *Automation*, *2*(4), pp.238-251.

[3] Khan, M.A., Khan, D.M.Z., Khan, H., Hanzla, M., Jamal, M., Khalid, T. and Ullah, N., 2020. Garbage collector robot. *Indian J. Sci. Technol*, *13*(20), pp.2065-2070.

[4] Rama Prabha, D., Mahindru, S., Srivastava, A. and Nilalohita, P., 2018. Autonomous Garbage Collector-Robodumpster. *International Journal of Civil Engineering and Technology (IJCIET)*, *9*(12), pp.545-552.

[5] Kulshreshtha, M., Chandra, S.S., Randhawa, P., Tsaramirsis, G., Khadidos, A. and Khadidos, A.O., 2021. OATCR: Outdoor autonomous trash-collecting robot design using YOLOv4-tiny. *Electronics*, *10*(18), p.2292.

[6] Kulshreshtha, M., Chandra, S.S., Randhawa, P., Tsaramirsis, G., Khadidos, A. and Khadidos, A.O., 2021. OATCR: Outdoor autonomous trash-collecting robot design using YOLOv4-tiny. *Electronics*, *10*(18), p.2292.

[7] Sunitha Devi, P., Suma, K., Srujana Reddy, K., Mandakini, C. and Pallavi Reddy, R., 2022, November. Robotic ARM for Effective Environmental Cleaning. In *International Conference on Intelligent Computing and Communication* (pp. 557-565). Singapore: Springer Nature Singapore.

[8] Vinodha, D., Sangeetha, J., Sherin, B.C. and Renukadevi, M., 2020. Smart garbage system with garbage separation using object detection. *International Journal of Research in Engineering, Science and Management*, *3*(5), pp.779-782.

[9] Hussain, S.B. and Kanwal, F., 2016, August. Design of a 3 DoF robotic arm. In *2016 Sixth International Conference on Innovative Computing Technology (INTECH)* (pp. 145-149). IEEE.

[10] Mohammed, A.A. and Sunar, M., 2015, May. Kinematics modeling of a 4-DOF robotic arm. In *2015 International Conference on Control, Automation and Robotics* (pp. 87-91). IEEE.

[11] Megalingam, R.K., Katta, N., Geesala, R., Yadav, P.K. and Rangaiah, R.C., 2018, December. Keyboard-based control and simulation of 6-DOF robotic arm using ROS. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)* (pp. 1-5). IEEE.
[12] Chikurtev, D., Rangelov, I., Chivarov, N., Markov, E. and Yovchev, K., 2018. Control of robotic arm manipulator using ros. *Bulgarian Academy of Sciences-Problems of Engineering Cybernetics and Robotics*, *69*, pp.52-61.

[13] Čermák, R., 2014. Web Camera Based Control of a Mitsubishi Melfa Robotic Arm with MATLAB Tools. *Technical Computing Bratislava*.

[14] Susanto, W., Rohrbach, M. and Schiele, B., 2012. 3d object detection with multiple kinects. In *Computer Vision–ECCV 2012. Workshops and Demonstrations: Florence, Italy, October 7-13, 2012, Proceedings, Part II 12* (pp. 93-102). Springer Berlin Heidelberg.

[15] B. Raafiu and P. A. Darwito, "Identification of Four Wheel Mobile Robot based on Parametric Modelling," 2018 International Seminar on Intelligent Technology and Its Applications (ISITIA), Bali, Indonesia, 2018, pp. 397-401, doi: 10.1109/ISITIA.2018.8710761.

[16]Tadic, V., Odry, A., Kecskes, I., Burkus, E., Kiraly, Z. and Odry, P., 2019. Application of Intel realsense cameras for depth image generation in robotics. *WSEAS Transac. Comput*, *18*, pp.2224-2872.

[17]Bobulski, J. and Kubanek, M., 2019. Waste classification system using image processing and convolutional neural networks. In *Advances in Computational Intelligence: 15th International Work-Conference on Artificial Neural Networks, IWANN 2019, Gran Canaria, Spain, June 12-14, 2019, Proceedings, Part II 15* (pp. 350-361). Springer International Publishing.

[18] Othman, N.A., Salur, M.U., Karakose, M. and Aydin, I., 2018, September. An embedded real-time object detection and measurement of its size. In *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)* (pp. 1-4). IEEE.

[19]Wang, C., Qin, J., Qu, C., Ran, X., Liu, C. and Chen, B., 2021. A smart municipal waste management system based on deep-learning and Internet of Things. *Waste Management*, *135*, pp.20-29.
[20]N. Genc, M. Nabi, H. Uzmus.December 2016. International Journal on "Technical and Physical Problems of Engineering" (IJTPE),Issue 29 ,Volume 8 Number 4, Pages 39-43.

[21]Chu, Y., Huang, C., Xie, X., Tan, B., Kamal, S. and Xiong, X., 2018. Multilayer hybrid deep-learning method for waste classification and recycling. *Computational intelligence and neuroscience*, *2018*.

[22]Bobulski, J. and Kubanek, M., 2021. Deep learning for plastic waste classification system. *Applied Computational Intelligence and Soft Computing*, *2021*, pp.1-7.

[23]Potrimba, P., 2023. What is a convolutional neural network. *Roboflow Blog*.

[24]Jacob Solawetz and Francesco. What is yolov8? the ultimate guide., 2023.
Website:
https://blog.roboflow.com/whats-newinyolov8/#:~:text=YOLOv8%20is%20the%20newest%20state,and%20industry%2Ddefining%20YOLOv5%20model.
Last Access Date : 1 jan, 2024

[25] Yolo8 vs Yolo5
Website: https://ar5iv.labs.arxiv.org/html/2305.09972
Last Access Date: 1 Jan, 2024