Department of Electrical and Computer Engineering
North South University

Senior Design Project

Autonomous Garbage Collector Rover Using Image Processing

Submitted by

MOIN SHAHRIYAR	**ID # 1610586042**
FARJANA ALAM	**ID # 1620060042**
SARHAN OSMAN BHUIYA	**ID # 1611008042**

Faculty Advisor:

Dr. Shazzad Hosain
Professor
Electrical and Computer Engineering Department

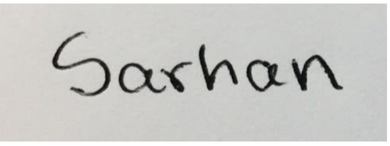
Fall, 2019

DECLARATION

This is to declare that no part of this report or the project has been previously submitted elsewhere for the fulfillment of any other degree or program. Proper acknowledgement has been provided for any material that has been taken from previously published sources in the bibliography section of this report.

....
MOIN SHAHRIYAR
ID-1610586042
ECE Department
North South University, Bangladesh

....
FARJANA ALAM
ID-1620060042
ECE Department
North South University, Bangladesh



....
SARHAN OSMAN BHUIYA
ID-1611008042
ECE Department
North South University, Bangladesh

APPROVAL

The Senior Design Project entitled "**Autonomous Garbage Collector Rover Using Image Processing**" by **MOIN SHAHRIYAR (ID-1610586042), FARJANA ALAM (ID-1620060042)** and **SARHAN OSMAN BHUIYA (ID-1611008042)** has been accepted as satisfactory and approved for partial fulfillment of the requirement of the Degree of Bachelor of Science in Computer Science and Engineering on the month of December, 2019. .

Supervisor's Signature

Md. Shazzad Hosain
Professor
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh.

Department Chair's Signature

Dr. K.M.A. Salam
Professor
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh.

ACKNOWLEDGMENT

First of all, we wish to express our gratitude to the Almighty for giving us the strength to perform our responsibilities and complete the report.

The senior design program is very helpful to bridge the gap between the theoretical knowledge and real life experience as part of Bachelor of Science (BSc) program. This report has been designed to have a practical experience through the theoretical understanding.

We also acknowledge our profound sense of gratitude to all the teachers who have been instrumental for providing us the technical knowledge and moral support to complete the project with full understanding.

It is imperative to show our appreciation for our honorable faculty member Dr. Shazzad Hosain for his undivided attention and help to achieve this milestone. Also, our gratefulness is divine to the North South University, ECE department for providing us a course such as CSE 499 in which we could really work on this project and materialize it the way we have dreamt of.

We thank our friends and family for their moral support to carve out this project and always offer their support.

ABSTRACT

We are presenting a smart robot that could present a viable solution towards efficient waste management which is based on embedded, digital image processing. The system is designed such that it can manually as well as automatically detect and collect the garbage. The detection is done by using the image processing algorithm SSD Mobilenet v2, which has been modified by adding our customized layer to the VGG-16 architecture for better detection. The real time video taken by the USB camera will be processed by the Raspberry Pi. For each frame taken it will detect any garbage present in the frame and categorize the detected garbage as Paper, Plastic, Metal or Glass. Once the sorting and categorization is complete the Raspberry Pi will send signals to the robot. Our robot will calculate its position, calibrate the motors according to the position of the garbage so that it will go to the acquired position and collect the garbage with help of customized robot arm. The pre-categorized collected garbage will be dumped in separate parts of a basket labeled as Paper, Plastic, Metal and Glass.

TABLE OF CONTENT

CHAPTER 5: USECASE/SUBSYSTEM

CHAPTER 6: SYSTEM DESIGN/ARCHITECTURE

CHAPTER 7: SYSTEM SOFTWARE

CHAPTER 8: ROBOT VEHICLE

CHAPTER 9: ROBOT ARM

CHAPTER 10: FEATURES

CHAPTER 11: DESIGN IMPACT

CHAPTER 12: RESULTS

CHAPTER 13: CONCLUSION

CHAPTER 14: APPENDIX

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

The main objective of this project is to recognize and categorize the waste autonomously, which require minimal human intervention. The robot will collect waste categorizing them as Plastic, Paper, Glass and Metal and put them in a container creating a cleaner and safer environment for humans to live.

At first, using the USB camera attached to the Raspberry Pi, our robot will try to find any garbage present in its surrounding location. Once the garbage is detected it will be classified into Paper, Plastic, Glass or Metal. The robot will go towards the garbage and if for any reason there is an obstacle present in-front of it, the garbage will avoid the obstacle using the help of sonar sensors attached to it. Once the robot is in-front of the garbage it will detect whether the categorized garbage is at a specific distance in-front of the robot using another sonar sensor. If the distance does not match, our robot will relocate its position and check the distance again. Once the distance matches, our robot will activate its robot arm and grab the garbage. After grabbing the garbage, the arm will drop it in the basket attached behind the robot. The basket is divided into 4 sections and is connected to a servo which helps it to rotate. Each section of the basket is labelled Paper, Plastic, Metal or Glass. Once the garbage is categorized the basket will rotate to that specific categorized part and the robot arm will drop the garbage in that specific section of the basket. All of this will happen automatically.

This system isolates waste automatically utilizing sensors, motors and detection software. The proposed system concentrates on identification, classification and segregation of waste. As the system works independently, there is no need of human mediation to control or to do any assignment. Utilizing Raspberry Pi and USB camera, the characterization result of images taken will automatically categorize the garbage as Plastic, Paper, Glass and Metal and send signals to the robot to collect the garbage. Once our modified robot gets the signal it will move towards the garbage and collect it using the modified robot arm. Finally the garbage will be dumped in separate portions of a basket based on the type of garbage collected. Our robot is different from other garbage collector robots as it can run in any unknown location. Sonar sensors allow it to avoid obstacles. Moreover, we used SSD Mobile Lite for detection of garbage which has lower processing power but great detection level. As a result, it will not heat up the processor of Raspberry Pi allowing it to run smoothly.

CHAPTER 1
(PROJECT OVERVIEW)

1.1 INTRODUCTION

We are presenting a smart robot that could present a viable solution towards efficient waste management which is based on embedded, digital image processing. The system is designed such that it can manually as well as automatically detect and collect the garbage. The detection is done by using the image processing algorithm SSD Mobilenet v2, which has been modified by adding our customized layer to the VGG-16 architecture for better detection. The real time video taken by the USB camera will be processed by the Raspberry Pi. For each frame taken it will detect any garbage present in the frame and categorize the detected garbage as Paper, Plastic, Metal or Glass. Once the sorting and categorization is complete the Raspberry Pi will send signals to the robot. Our robot will calculate its position, calibrate the motors according to the position of the garbage so that it will go to the acquired position and collect the garbage with help of customized robot arm. The pre-categorized collected garbage will be dumped in separate parts of a basket labeled as Paper, Plastic, Metal and Glass.

1.2 PROBLEM DESCRIPTION

Bangladesh is the ninth most populous and twelfth most densely populated country in the world. With this population growth, there is an increasing problem of waste management particularly in the larger cities. Currently, according to an UNFPA report, Dhaka is one of the most polluted cities in the world and one of the issues concerned is the management of municipal waste. Current (2019) waste generation in Bangladesh is around 22.4 million tons per year or 150 kg/cap/year.[1] There is an increasing rate of waste generation in Bangladesh and it is projected to reach 47,064 tons per day by 2025. The rate of waste generation is expected to increase to 220 kg/cap/year in 2025. A significant percentage of the population has zero access to proper waste disposal services, which will in effect lead to the problem of waste mismanagement. The total waste collection rate in major cities of Bangladesh such as Dhaka is only 37%. When waste is not properly collected, it will be illegally disposed of and this will pose serious environmental and health hazards to the Bangladeshis. [2]

One of the most adverse impacts of poor waste management, especially municipal waste, is the incidence and prevalence of diseases such as malaria and respiratory problems, as well as other illnesses through the contamination of ground water. Biomedical wastes pose great danger in Bangladesh too as a report estimated that 20% of the biomedical waste is "highly infectious" and is a hazard since it is often disposed of into the sewage system or drains.[3] With regards to the living standards, solid waste leads to blockage in the drainage system which leads to flooding in the streets. Consequently, mosquitoes and bad odor are among the negative impacts resulted. [4] The main objective of this project is to recognize and categorize the waste autonomously, which require minimal human intervention. The robot will collect waste categorizing them as Plastic, Paper, Glass and Metal and put them in a container.

1.3 PROJECT DETAILS

We have a smart robot that can collect and categorize waste or garbage in Paper, Plastic, Metal and Glass and collect them in a container attached to the back of the robot. At first, using the USB camera attached to the Raspberry Pi, our robot will try to find any garbage present in its surrounding location. Once the garbage is detected it will be classified into Paper, Plastic, Glass or Metal. The robot will go towards the garbage and if for any reason there is an obstacle present in-front of it, the garbage will avoid the obstacle using the help of sonar sensors attached to it. Once the robot is in-front of the garbage it will detect whether the categorized garbage is at a specific distance in-front of the robot using another sonar sensor. Once the distance matches, our robot will activate its robot arm and grab the garbage. After grabbing the garbage, the arm will drop it in the basket attached behind the robot. The basket is divided into 4 sections and is connected to a servo which helps it to rotate. Each section of the basket is labelled Paper, Plastic, Metal or Glass. Once the garbage is categorized the basket will rotate to that specific categorized part and the robot arm will drop the garbage in that specific section of the basket. All of this will happen automatically.

1.4 PROJECT GOAL

The goal of the proposed system is to concentrate on identification, classification and segregation of waste. The system will be able to isolate waste automatically utilizing sensors, motors and detection software. As the system works independently, there is no need of human mediation to control or to do any assignment. Our robot will be different from other garbage collector robots as it can run in any unknown location. We will use SSD Mobilenet v2 model for detection of garbage, which is based on VGG16 Convolutional Neural Network architecture. Moreover we can improve our detection accuracy by including our customized convolution layers in this model. Sonar sensors will allow it to avoid obstacles. As a result, it will not heat up the processor of Raspberry Pi allowing it to run smoothly. We will also added high power Denso Motors which allow the robot vehicle to easily overcome medium sized rocks. Our robot will have a rotating basket divided into four parts (Paper, Plastic, Metal and Glass) attached to a servo. So, if paper is collected the basket will rotate to the side of the paper and the robotic arm will drop the paper in the specific paper part of the basket.

1.5 PROPOSED SOLUTION

The proposed system aims to recognize and categorize the waste autonomously, which require minimal human intervention. This entire process of recognition of waste material is based on the detection of the garbage. The system will be trained through datasets by using machine learning technique such as SINGLE SHOT DETECTION (SSD). Utilizing Raspberry Pi the characterization result will be given to the equipment part of the framework with the goal that it will be dumped in its separate containers. The system will collect waste automatic categorizing them as Plastic, Paper, Glass and Metal. As the system works independently, there is no need of human mediation to control or to do any assignment. The collected garbage will be specified as Paper, Plastic, Metal and Glass and kept in different parts of a basket attached to the back of our robot.

CHAPTER 2
(LITERATURE REVIEW)

There are numerous methods that are used to make a garbage collecting robot. These methods provide the information about various techniques used for a garbage collecting robot. Some of these methods do have limitations which are proposed in this section.

2.1 THOOYAN

A cost effective garbage cleaning robot is developed and that is named as “Thooyan” [1]. The system (road cleaning robot) consists of very simple but highly efficient mechanism. The main components consist of a rotating brush assembly (rake), a unique tilting wedge, a conveyor system and a garbage collection unit. Robot is programmed in a certain pattern so as to navigate automatically and detect obstacles to move in a free path. If encountered by a moving obstacle, the robot is programmed to pause for duration of 50 seconds and then sense again to move or it will take turn of 180 degree. A solar panel is provided for partial charging of the battery. Since this robot uses conveyor belt the cost of the whole system will be more which adds limitation to this method. But it gave an idea or advantage to use solar panels which in turn helps to reduce the power consumption. This robot is the small step to change the manual waste collection and ensures the safety of sanitary workers. The limitation of this robot is that it can only run on free paths, where else our robot can run in any unknown location.

2.2 ROAR

The robot, called ROAR[2] is transported to the refuse collection site on the back of a refuse truck. An operator presses a button on the truck and this prompts a drone to be launched from its roof and begin scanning the surrounding area to locate bins. The locations of the bins are then relayed to the ground-based robot. ROAR navigates its way to each bin using a map of the area and the likely locations of bins, as well as the data provided by the drone. GPS and LiDAR are used to help it navigate and avoid obstacles. Inertial measurement unit (IMU) data, from accelerometer and gyroscope sensors, are used to help the robot keep track of its position. Once ROAR has arrived at the bin, it uses cameras and LiDAR to position itself, before extending its arms and lifting the bin onto its built-in platform. It then returns to the refuse truck and lifts the bin into position to be emptied. Manual intervention is needed which is the demerit. This robot uses GPS to avoid obstacles where else our robot does not need GPS to navigate. It can function in any unknown location.

2.3 ROBODUMPSTERA

Another robot called ROBODUMPSTERA [3] has a robotic arm which has an anatomy similar to the human arm. They are becoming very popular as research platforms. Progress of robotic arm is inhibited due to a shortage of affordable platforms with wide capabilities. In this work, we present infrared based robotic arm. This robotic arm has been developed with sufficient power and capabilities so as to be employed for various applications. The function which has been implemented is of an autonomous garbage collector. It measures breadth of the object and correspondingly grab or avoid it. As a result it can only grab specific objects which match the required breadth.

2.4 DUSTCART

The design and the experimental results of DustCart [4], a wheeled autonomous robot for door-to-door garbage collection. DustCart is able to navigate in urban environments avoiding static and dynamic obstacles and to interact with human users. The robot is managed by an Ambient Intelligence system (AmI) through a wireless connection: it navigates to collect garbage bags to the houses of users and then moves to discharge the collected waste to a dedicated area. The architecture, navigation and localization systems are described along with the results achieved in different urban sites. In particular, a localization approach based on optical beacons was used and guaranteed position errors sufficient for a safe robot navigation. We report also the first results of a long-term experimentation of the DustCart \cite{b4} robot in Peccioli, a small town of Tuscany (Italy). This can be considered for only limited small locations. Our robot can function in every unknown location as it has a custom built object avoidance program using sonar sensors.

2.5 COMPARISON

In all, our robot is different from others because it has a high power motors which allows it to move over almost any medium sized stones or rocks. It works in both autonomous and human control modes. It has a custom built obstacle avoidance program which allows it to move in any unknown location. It has a GPS so that if there is any attempt of theft immediate actions can be taken. As we used image processing, the garbage is categorized into Paper, Plastic, Metal and Glass. It has a custom built robot arm that can carry garbage up to 1kg and place it in different sections of the basket attached behind it. As a result, there is no need of any categorization by a human after the garbage has been collected. Lastly, the main objective of this project is to recognize and categorize the waste autonomously, which require minimal human intervention. The robot will collect waste categorizing them as Plastic, Paper, Glass and Metal and put them in a container creating a cleaner and safer environment for humans to live.

CHAPTER 3

SYSTEM DESIGN

Our project is basically the idea of detection and collection. The detection part will be done by the Raspberry Pi and the collection of garbage will be done by the robot. As a result, we have divided our project into two major parts.

3.1 DETECTION

The system will be trained through a dataset by using machine learning technique such as SINGLE SHOT DETECTION (SSD). We developed our dataset which consists a total of 2518 RGB images for both training and testing images with JPG format. We hand collected the dataset as well as used an existing dataset from Kaggle. We have taken images from different angles and lights so that our machine can learn variation and angles of the garbage. We re-sized those images into 512*384 pixels for training model to run without any error and delay. Our images has been labelled by labelImg application for supervised learning into four classes (paper, plastic, glass and metal). After la-belling, our data is ready to train and test. We used 80 percent of the dataset to train and 20 percent to test. Finally we implemented the developed dataset in a Raspberry Pi which helps our robot to collect waste and automatically categorizing them as Plastic, Paper, Glass and Metal. As the system works independently, there is no need of human mediation to control it. As we are using a USB camera, our robot can detect a garbage within 1 feet of the robot. But if we switch to a better camera, the range of detection will significantly increase.

3.2 COLLECTION

For the collection part, the robot will at first go in front of the garbage and with the help of sonar sensors will calculate the exact distance between the garbage and the robot. Later it will send the data to the Arduino which will decide if it is possible to collect the garbage or not within that calculated distance. If it is possible, Arduino will send a signal to the robotic arm to grab the waste in front of it. Once the garbage is collected, it will drop the garbage in a basket which will be divided into 4 parts (Paper/Plastic/Metal/Glass). The basket will be rotating based on which type of garbage has been collected. So, if paper is collected the basket will rotate to the side of the paper and the robotic arm will drop the paper in the specific paper part of the basket.

3.3 WORKING PROCEDURE

At first our robot will detect a garbage with the help of trained model in Raspberry pi. After processing the data, it will send signals to the robot to go to the specific location of the garbage and finally collect it in a specific part of the basket or container. At first, using the USB camera attached to the Raspberry Pi, our robot will try to find any garbage present in its surrounding location. Once the garbage is detected it will be classified into Paper, Plastic, Glass or Metal. The robot will go towards the garbage and if for any reason there is an obstacle present in-front of it, the garbage will avoid the obstacle using the help of sonar sensors attached to it. Once the robot is in-front of the garbage it will detect whether the categorized garbage is at a specific distance in-front of the robot using another sonar sensor. If the distance does not match, our robot will relocate its position and check the distance again. Once the distance matches, our robot will activate its robot arm and grab the garbage.

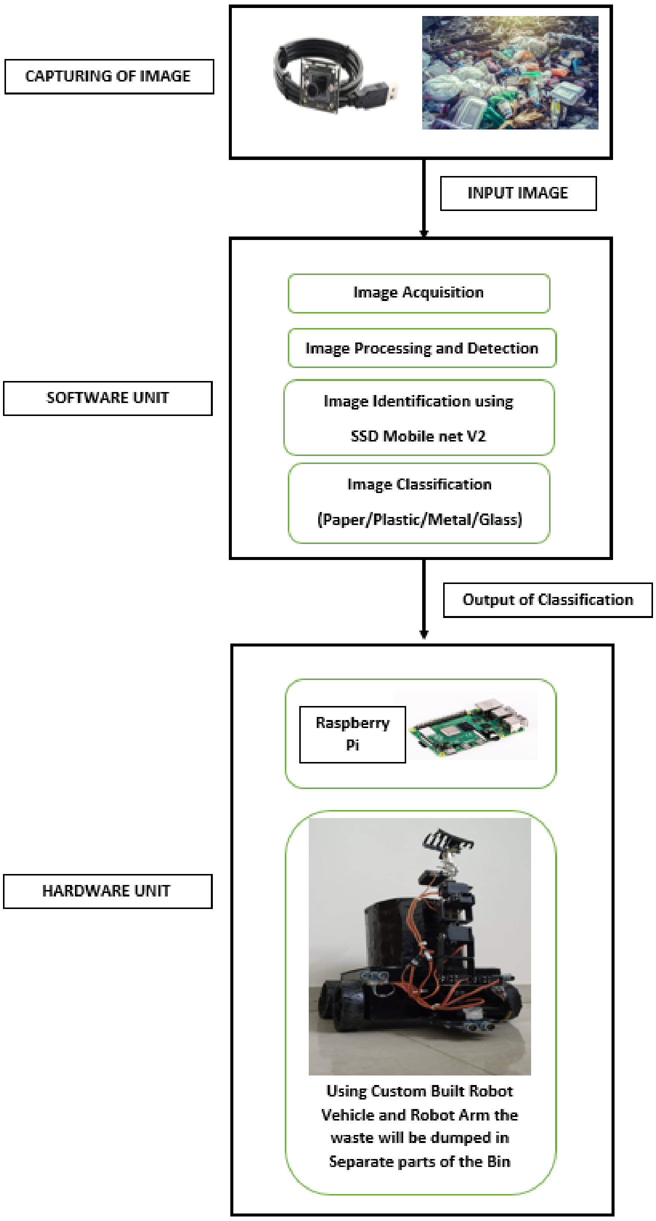


Fig. 3.1. Process of Autonomous Garbage Collection

After grabbing the garbage, the arm will drop it in the basket attached behind the robot. The basket is divided into 4 sections and is connected to a servo which helps it to rotate. Each section of the basket is labelled Paper, Plastic, Metal or Glass. Once the garbage is categorized the basket will rotate to that specific categorized part and the robot arm will drop the garbage in that specific section of the basket. All of this will happen automatically.

CHAPTER 4
(COMPONENT DETAILS)

4.1 ROBOT VEHICLE COMPONENTS

The robot vehicle is considered to be the main part of the robot. It has customized metal body of Length: 30 cm, Width: 26 cm, Height: 13 cm and Wheel Diameter: 11cm. It is 2 cm above the ground level. All the important devices like Raspberry Pi, Arduino Mega, etc. has been kept inside this vehicle's body. The body contains an extended part in the front to attach the robot arm which will collect the garbage. It has a torque power of 10kg per square cm.

Toyota Denso Window Motor

RPM : 80±10rpm (no load)
55±15rpm (with load)
Max Voltage: 12v
No Load Current: < 2A
Stall Current: < 8A
Rated Torque: 3 Nm ~ 30.6 kg-cm
Stall Torque: 9.8N.m ~ 100kg-cm

Fig. 4.1. Toyota Denso Window Motor

4.2 ROBOT ARM COMPONENTS (MODIFIED):

We have run some experiment regarding the grabbing part of the robotic arm. The most available servo motor in Bangladesh is MG996R / MG995. Using this servo, it is quite difficult to grab the object from the ground using 6DOF arm. Though we bought 6 DOF arm but we modified the structure of the arm so that it can grab the objects so frequently.

Item Name	Weight (gm)	Length (cm)	Image
1 x Metallic Claw	56	11	
3 x Long U Shaped Bracket	22	6.4	

3 x Multifunctional Bracket	**16**	**5.6**	
1 x Cup Bearing	**-**	**-**	
Item Name	**Weight (gm)**	**Length (cm)**	**Image**
2 x Hard U Beam	**50**	**9**	
L-Bracket	**6**	**-**	
7 x MG996R Servo	**55**	**~4**	

Fig. 4.2. Robot Arm Components List

4.3 OVERALL COMPONENTS:

The table below comprises of the major components required to build our robot. The component and its quantity might change later as required for the project.

NUMBER	COMPONENTS	QUANTITY
1	Customized Metallic Robot Body	1
2	Toyota Denso Motor	4
3	12V 7.5 AH Power Supply	1
4	4-Channel Relay	2
5	Arduino Mega	1
6	Node MCU	1
7	Robotic Hand 6DOF	1
8	Mg996R Servo	8
9	Metal Horn	8
10	5G90S Servo	1
11	6V 4.5 AH Power Supply	1
12	Sonar Sensor	3
13	Wheel 4 Inch	4
14	Raspberry Pi with 5MP Camera	1
15	Custom Plate to carry two servo in shoulder	1
16	Bread Board	1
17	5V Power Supply	1
18	Multi-Functional Bracket	1
19	Basket	1
20	Wires	(as required)

Fig. 4.3. Overall Components List

CHAPTER 5
(USECASE/SUBSYSTEM)

There are basically two use case for this project. One of them is autonomous and the other is human control.

5.1 USECASE-1 (AUTONOMOUS)

Here the Garbage Collector Robot will be a driverless autonomous vehicle specifically designed to collect waste which require no human intervention. It will automatically detect the waste, move towards it and grab the waste with its robotic arm. Once the waste is collected, it will automatically drop the waste in its specified part (Paper/Plastic/Metal/Glass) of the container.

5.2 USECASE-2 (HUMAN CONTROL)

Here the Garbage Collector Robot can be controlled manually by a human with the help of an app. The robot can be manually sent front, back, right and left using button controls on the app and can also pickup any waste from the ground using the robotic arm.

5.3 USER MANUAL

Our robot is based on specific conditions like Human Control or Autonomous. A diagram related to its working principle has been generated.

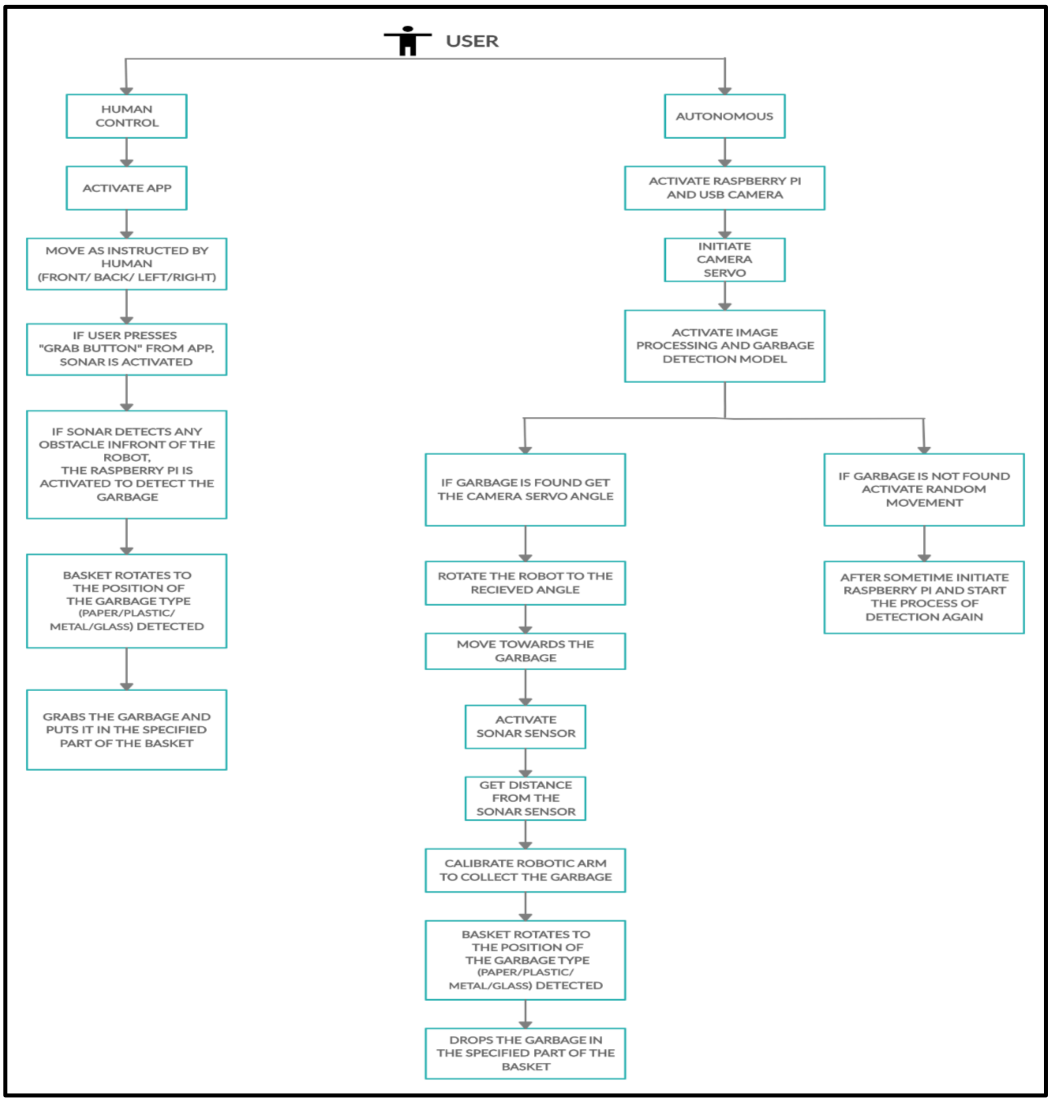


Fig. 5.1. User Manual for Autonomous and Human Control Garbage Collector Rover

CHAPTER 6
(SYSTEM DESIGN / ARCHITECTURE)

6.1 DESIGN

The robot is basically of 4 main parts. At first for the software i.e. the detection part, we used Raspberry Pi which was trained to detect wastes like Paper, Plastic, Metal and Glass. Next, for the movement of the robot we used 4WD robot vehicle that allows the robot to move in any direction it wants to. Thirdly, we attached a Robot Arm to the vehicle which gradually grabs the garbage it detects. Finally, the robot has a rotating bin/basket behind it which collects the garbage grabbed by the robotic arm. So, if paper is collected the basket will rotate to the side of the paper portion of the container and the robot arm will drop the paper in the specific paper part of the basket. As a result, Paper, Plastic, Metal and Glass will be collected separately in different portions of the basket.

6.2 ARCHITECTURE

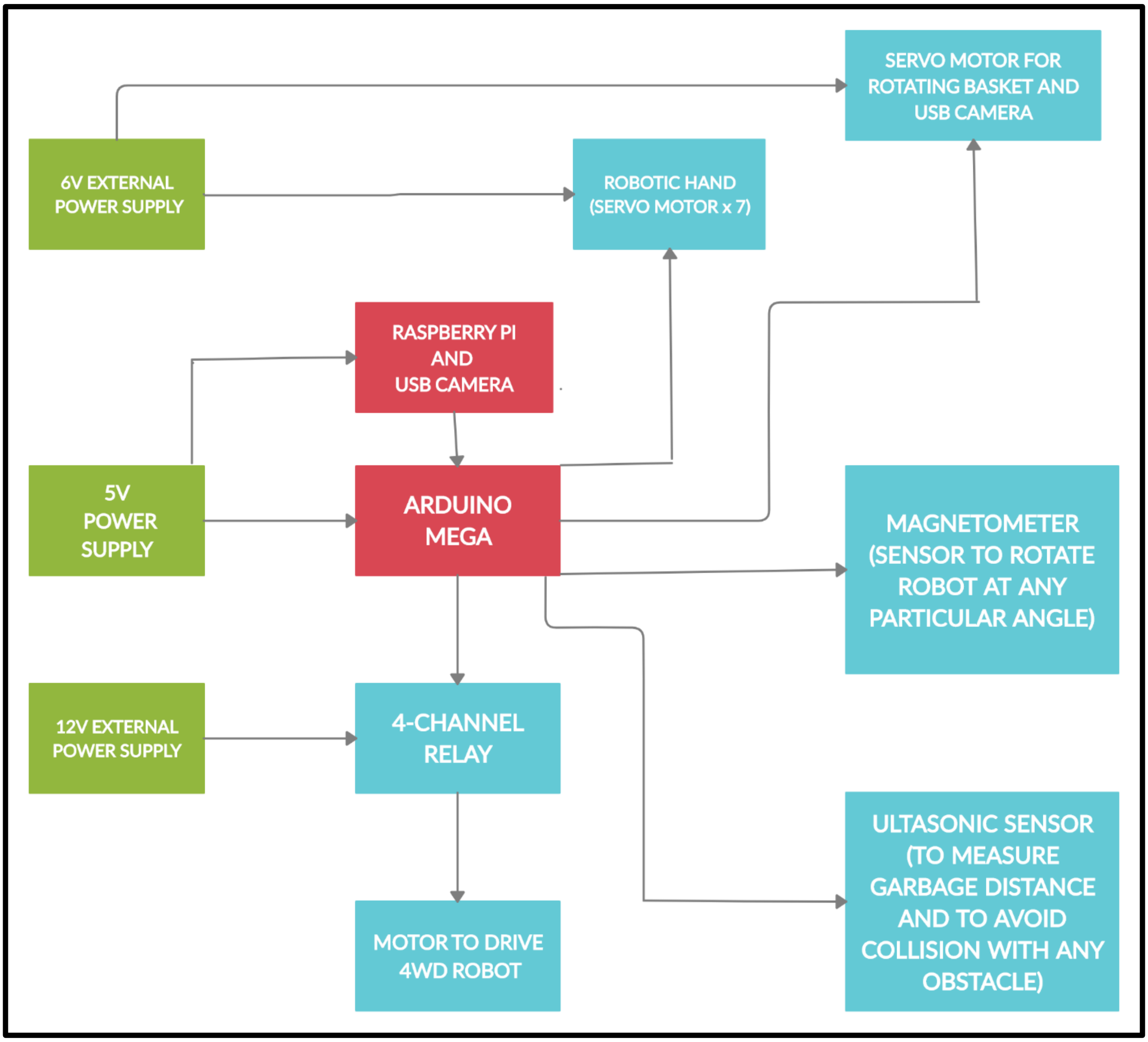


Fig. 6.1. Architecture of Autonomous Garbage Collection Rover

CHAPTER 7
(SYSTEM SOFTWARE)

In the field of computer vision, convolution neural networks excel at image classification, which consists of categorizing images, given a set of classes and having the network determine the strongest class present in the image.

7.1 SINGLE SHOT

This means that the tasks of object localization and classification are done in a single forward pass of the network. SSD's architecture builds on the venerable VGG-16\cite{b6} architecture, but discards the fully connected layers. The reason VGG-16 was used as the base network is because of its strong performance in high quality image classification tasks and its popularity for problems where transfer learning helps in improving results. Instead of the original VGG fully connected layers, a set of auxiliary convolutional layers were added, thus enabling to extract features at multiple scales and progressively decrease the size of the input to each subsequent layer.

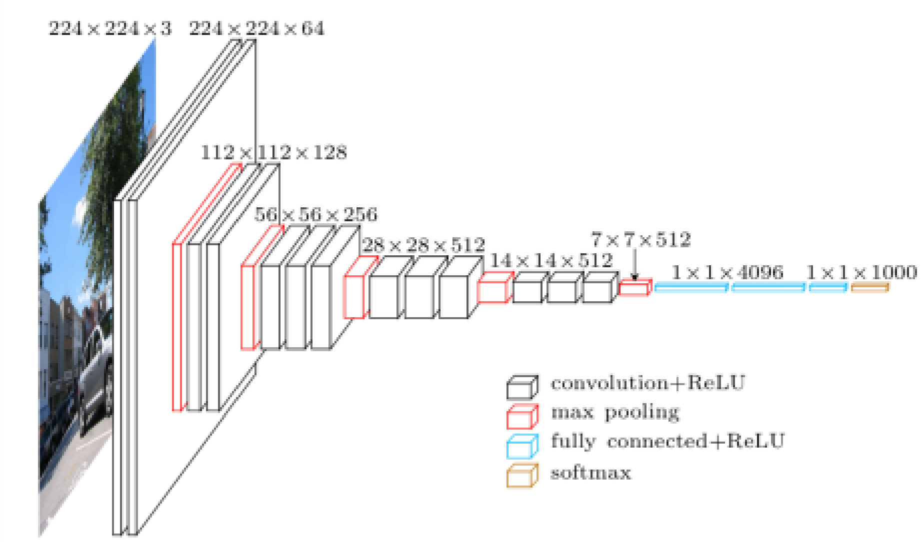


Fig. 7.1. VGG Architecture

7.2 MULTIBOX

MultiBox starts with the priors as predictions and attempt to regress closer to the ground truth bounding boxes. More default boxes results in more accurate detection, although there is an impact on speed. Having MultiBox on multiple layers results in better detection as well, due to the detector running on features at multiple resolutions.

CHAPTER 8
(ROBOT VEHICLE)

8.1 STRUCTURE OF ROBOT VEHICLE

The robot vehicle is considered to be the main part of the robot. It has customized metal body of Length: 30 cm, Width: 26 cm, Height: 13 cm and Wheel Diameter: 11cm. It is 2 cm above the ground level. All the important devices like Raspberry Pi, Arduino Mega, etc. has been kept inside this vehicle's body. The body contains an extended part in the front to attach the robot arm which will collect the garbage. It has a torque power of 10kg per square cm.

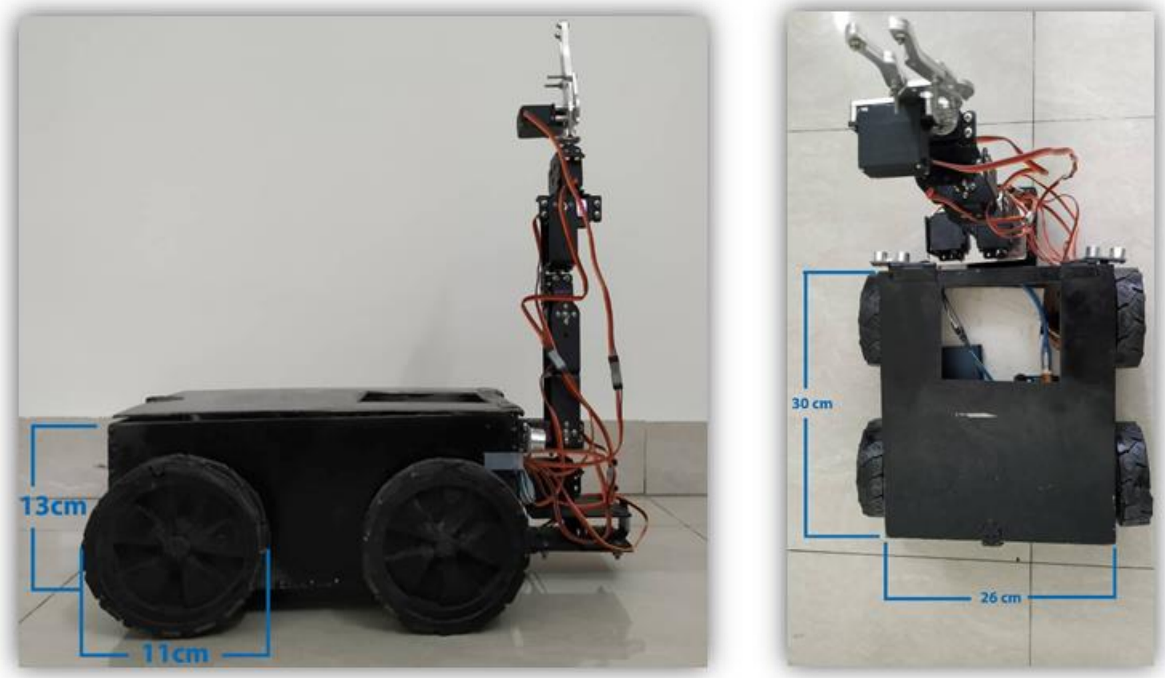


Fig. 8.1. Structure of Robot Vehicle

8.2 TOYOTA DENSO WINDOW MOTOR

RPM : 80±10rpm (no load)
55±15rpm (with load)
Max Voltage: 12v
No Load Current: < 2A
Stall Current: < 8A
Rated Torque: 3 Nm ~ 30.6 kg-cm
Stall Torque: 9.8N.m ~ 100kg-cm

Fig. 8.2. Toyota Denso Window Motor

8.3 CALCULATION

$\vec{\tau} = \vec{l} \times \vec{F} = Fl \sin\theta\hat{n}$

While robot is moving forward:
Using right hand rule,

Direction of F is to the left creating 0-degree angle with the width for Right-sided motors.

Direction of F is to the left creating 180-degree angle with the width for Left-sided motors.

So, in both cases, length of width is canceled out as sin(0)=0.
But, Direction of F is to the left creating 90-degree angle with the length for Right-sided motors.

Direction of F is to the left creating 90-degree angle with the width for Left-sided motors.

So, for both cases, torque value should me maximum as sin(90)=1.
1 Denso motor gives torque of 100 kg-cm.
Length of the robot: 30cm
So, Maximum weight 1 motor can lift: 100/30 kg =3.3 kg
Maximum weight 4 motor can lift: 3.3kg x 4 = 13.33 kg
Same mechanism for moving backward, right and left.

Note: We have tested with 15kg weight above the robot and it worked absolutely fine. So, we can say the motors are much more powerful than it was scripted by the company.

8.4 POWER CONSUMPTION

One motor consumes approximately 2~2.5A depending on its movement. For instance, while moving forward and backward it consumes 2A. On the other hand, while moving right and left it consumes 2.5A. It is because of the drifting of both motor at the opposite directions. While moving right, left sided motors rotate clockwise and right sided motors rotate counter clockwise. To cancel out this opposite direction drifting each motors consumes higher current than moving forward or backward.

So, In total 4 motors consumes: (2~2.5 x 4) A = 8 ~ 10 A
We are using 12v 7.5Ah battery. So, the uptime of our robot is: 7.5/10 hr = 0.75 hr = 45 mins
It means, Our robot can be driven up to 45mins straight with full load and working capacity.
As our battery takes 2A initial current for charging it will take (7.5/2) hr = 3.75 hr = 3hr 45 mins to fully charged.

CHAPTER 9

(ROBOT ARM)

9.1 STRUCTURE OF ROBOT ARM

We have run some experiment regarding the grabbing part of the robotic arm. The most available servo motor in Bangladesh is MG996R / MG995. Using this servo, it is quite difficult to grab the object from the ground using 6DOF arm. Though we bought 6 DOF arm but we modified the structure of the arm so that it can grab the objects so frequently.

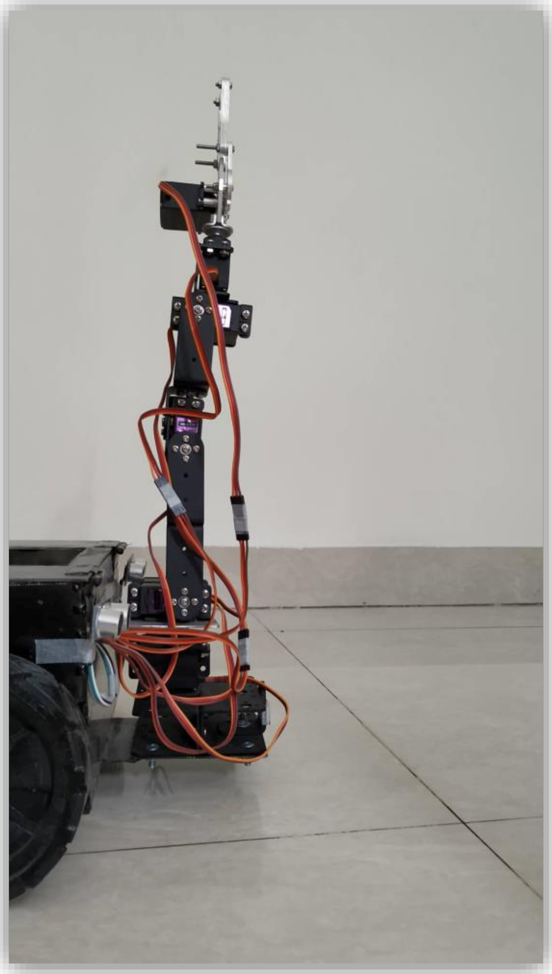


Fig 9.1: Robotic Arm (Modified)

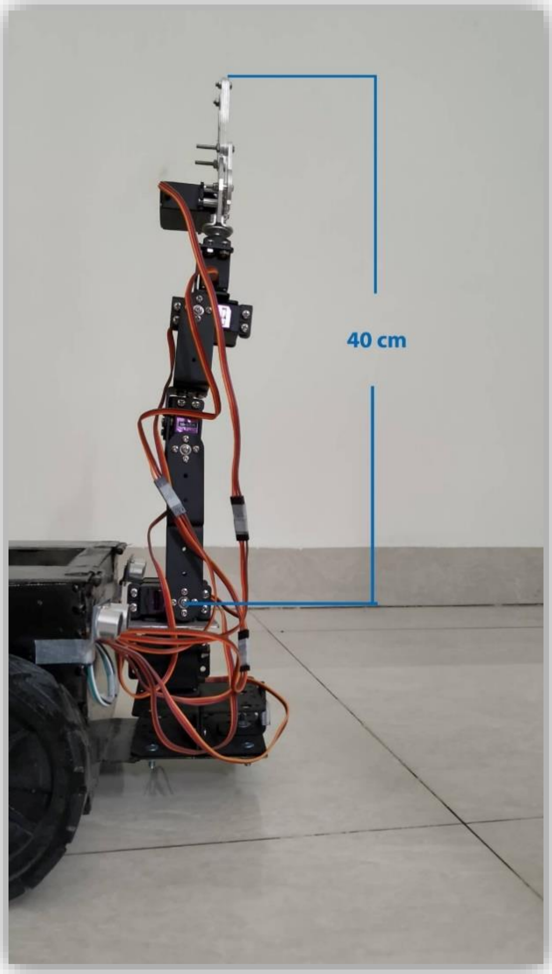


Fig 9.2: Robotic Arm (with required length)

9.2 CALCULATION

Torque of MG996R: **9.4 kg-cm** (at 4.8 v)
Torque of MG996R: **11 kg-cm** (at 6 v)
Operating Voltage: 4.8 v ~ 6.6 v
We are operating at 6 v using 6v 4.5Ah battery. So, we are getting the maximum torque of 11kg-cm.

9.3 TORQUE EQUATION

$\vec{\tau} = \vec{l} \times \vec{F} = Fl \sin\theta \hat{n}$

The length from Shoulder (Servo2) to the gripper top is: ~40 cm
Maximum weight 1 MG996R servo can lift: 11 kg-cm / 40 cm = 275 gm
So, Total weight shoulder servos can lift: 275 x 2 gm = 550 gm
Equipment's total weight above Servo2: Weight of Claw + Weight of U Bracket x 3 + MG996R servo x 4 + Multifunctional Bracket x 3 + L-type Bracket x1
= (56 + 22 x 3 + 55 x 4 + 16 x 3 + 6) gm
= 396 gm, which is greater than 275 gm but less than 550gm
So, Shoulder servos can lift total weight of garbage = (550 – 396) gm = 104 gm which is enough for our project.

9.4 MODIFICATION NECESSITY

If we used 1 servo in shoulder of the 6 DOF then the height from shoulder to the gripper top would be still approximately 40 cm.
Then it can lift maximum weight of = 11 kg-cm / 40 cm =0.275 kg ~ 275 gm
But component's total weight above Shoulder would be = (56 + 22 x 3 + 55 x 4 + 16 x 3 + 6) gm
= 396 gm which is greater than 275 gm
So, it will not work properly.
<u>Note:</u> In fact, we ran some tests earlier on 6DOF and we saw that Shoulder servo fall down with the excessive weight as it's torque is not compatible with that much weight.
So, we modified and used two servos together in the shoulder with the help of a customized metallic plate. After modifying our hand can grab the garbage and lift if properly. As our robotic hand is compatible with XYZ (3D) rotation so it is enough for our project.

9.5 POWER CONSUMPTIONS

We have used 7 servos for robotic hand, 1 servo to rotate the basket, 1 for rotating camera. So in total 9 servos are used.
Each servo has a stall current of 1.4 A.
So total Stall current: (9 x 1.4) A = 12.6 A
We are using 6v 4.5Ah for powering up robotic hand servos along with basket servo and camera servo. So the uptime of robotic hand and basket and camera rotation function is: (4.5/12.6) hour = 0.38 hour ~ 22 mins. This is the uptime with full load functionalities. All the servos will not work at full load capacity because of their position. Neither all the servos are at the maximum distance from the gripper of the robotic hand nor do they lift the maximum weight while picking garbage. So power (current) consumption is not equal for all the servos. We noticed that the actual uptime is almost triple of this time which is more than enough for our project.
The idle current draw for each servo: 10mA which is so low so we can ignore it.

CHAPTER 10 (FEATURES)

10.1 OBSTACLE AVOIDANCE

There are 3 sonar sensors in the front and 1 at the back of the robot vehicle to avoid obstacles in the pathway of the robot. The sonars in the front can rotate from 0 to 90 degrees with the help of the servo which helps the sonar sensors to increase the range of coverage on the left and right side of the robot. IR sensors are also used to put firm in the obstacle avoidance functionality. There is a IR sensor on the robotic arm at height 35cm just below the claw. This sonar helps to avoid any obstacle present in the range of the height of the robot arm.

10.2 LIVING BEINGS AVOIDANCE

There is a PIR motion sensor in the front of the robot. A passive infrared sensor (PIR) is an electronic sensor that measures infrared light radiating from objects in its field of view. It helps to detect any kind of movement in front of the robot and avoid it. A Thermal sensor (OMRON MEMS) is combined with PIR sensor in order to detect living being more precisely.

Future Improvement: "Detect living being from webcam through image processing."

10.3 MOVEMENT ACCURACY

The robot is based on image processing. So it will get a direction from the camera servo. Initially the camera servo starts rotating and if it detects any garbage nearby it will send the camera servo angle to the Arduino. Depending on the motor rpm robot will get an initial rotation time to rotate in that direction given by camera servo. While rotating the MPU6050 sensor (gyroscope, accelerometer) will confirm the rotation. As the robot moves forward both IR and Sonar sensor will confirm the remaining distance of the garbage from the robot.

10.4 SECURITY

A GPS is attached to the rover so that we can identify its location through google map. It is very important as any autonomous robot can be misused and stolen if we don't use GPS system. For example: If anyone picks up the robot and takes it out from it designated coordinates, a notification will be sent to the control. Besides, we can also detect whether the robot is working properly or not by seeing its location. For example: During autonomous mode if the robot is stuck in a specific location for more than 2 minutes, a notification will be automatically sent to the control room.

10.5 LIVE FEEDBACK

If there will be any sort of misbehavior from the robot the system will identify it and sends a message to the control room with the help of Arduino and SIM900A module. A sim card will be inside the module and it will send the error message to the control room.

10.6 HIGH POWER MOTORS

High Power Motors which helps to move over small or medium sized objects.
Maximum weight 4 motor can lift: 3.3kg x 4 = 13.33 kg. Since out robot is 6kg, it can easily move around its environment without any difficulties.

10.7 FUTURE IMPLEMENTATIONS

- Using Reinforcement Machine Learning Algorithm the robot will be taught the surrounding environment and will make its own decision whether to move next after grabbing one garbage. Obviously there will be an increase number of sensors to support this mechanism.

- Add solar chargers to the robot so that it can automatically charge itself when it runs out of power.

- Improve dataset by adding more images to it. This will increase the detection and accuracy level to a great extent.

CHAPTER 11
(DESIGN IMPACT)

11.1 ECONOMIC IMPACT

Waste is a piece of the economy. It is a side-effect of monetary movement, by organizations, government and family units. Waste is likewise a contribution to financial action, regardless of whether through material or energy recovery. The management of that waste has economic implications for efficiency, government use, and, obviously the environment. Since our project is a waste collecting robot that collects Paper, Plastic, Metal and Glass and keeps then in separate containers, it can play a vital role in the economic sector. It reduces local authority waste management budget due to decreased quantities of waste. The recycling of waste products and things that are not usable anymore is an extremely important process providing us with huge benefits. Glass, paper, metal and plastic can be recycled which will eventually put less pressure on precious natural resources thus reducing the cost of the production of many products. Collected waste can be recycled and used for other business purposes. With increased regulations aiming to reduce the danger of cutting down trees, recycling paper offers a wonderful business opportunity for someone looking to establish a profitable small business. Plastic recycling refers to the process of recovering waste or scrap plastic and reprocessing the materials into functional and useful products. This activity is known as the plastic recycling process. The goal of recycling plastic is to reduce high rates of plastic pollution while putting less pressure on virgin materials to produce brand new plastic products. This approach helps to conserve resources and diverts plastics from landfills or unintended destinations such as oceans. Most of the metal and glass wastes are collected by the same company of the waste to be recycled and used again. This reduces the cost leading to a greater amount of profit. Recycling of one glass container saves enough energy to light a 100-watt bulb for 4 hours.

- The economic cost of waste imposes costs to existing and potential future generations. Any process or product that produces waste that cannot be assimilated back into the environment safely should not be permitted.
- Financial benefits to business through reduced expenditure on waste disposal, but also through more intelligent purchasing.
- Opportunity costs of clean up campaigns and behavior change initiatives.
- Wastage sorting is a huge task to do. Many companies use thousands of dollars on different procedures to sort the wastage into categories like plastic, paper etc. as it is a required to recycle them. Our robot can sort the garbage while picking it, as a result there is no need of sorting the garbage after collecting it. It can save both money and energy.

11.2 ENVIORNMENTAL AND SOCIAL IMPACT

Our project helps to collect waste which can turn unwanted waste into useful substances such as compost and waste energy. We can also help to reduce the amount of greenhouse emissions and leachate production. Finally, we can help conserve space in landfills and also natural resources such as water, timber and minerals, which would otherwise be used in the manufacturing of new materials.

- Further research into incineration and energy recovery can be undertaken.
- Risk of contamination to ground water systems reduces
- Dust and litter reduce to surrounding areas
- Reduce demand for landfill sites competes with more sustainable land uses.
- Further research into CO2 and methane emissions associated with landfill and transport is reduced
- Minimizes greenhouse gas emissions associated with waste collection, transportation and treatment
- Keeps the environment clean and fresh
- Saves the Earth and conserves energy
- Reduces environmental pollution
- Conserves the beauty of nature and landscapes
- A spectacular improvement on tourists sites by keeping them clean.

11.3 HEALTH BENEFITS

The most important benefit of waste collection with our robot is the protection of nature and health of the entire living population. Rubbish and waste is lying in the open can cause air, water, and land pollution. The inhaling of the polluted air can damage the respiratory system and cause nausea. The consuming of water that is contaminated with hazards of the rotting garbage can cause various diseases such as cholera. That's why water sources and air must be protected at all cost, and the best way to protect them is to collect trash. In addition to that, as we all know plastic decreases the fertility of the soil. Our project collects plastic from the ground which prevents plastic to get mixed with soil. This results to improve the health of the soil gradually improving the health of trees. Metal is another issue related to soil health which can be solved with the help of our project. If our environment is healthy, so are we. This is how our project has a great impact over health.

11.4 SAFETY CONSIDERATIONS

First of all, our robot will be able to detect any presence of humans or any moving object. It will stop and stay still in one position until there is no moving obstacle or person in front of it. So, if any kid or person is walking in the park or the area where the robot will be deployed, it will be a very safe condition for the human as well as the robot.

Secondly, if our robot crashes or gets stuck in one position for a long time, it will automatically send signals to the owner so that it can be corrected. Signals will be sent based on co-ordinates received from the GPS allocated within the robot.

Thirdly, if the robot hand is unable to collect the garbage it will check all the servos position of the robotic hand. If any of the servos are responsible for the hand to get stuck it will try to reset the hand. If the resetting process fails, the robot will alert the authority and sends it coordinates so that immediate care can be taken.

Fourthly, our robot has a feature of obstacle avoidance. So, while it is moving autonomously it can avoid collision with any sort of obstacle such as wall, bench, tree etc. Additionally, if it detects any obstacle in front of any garbage that is detected by it, it can change position and try from a different angle to pick up the garbage.

Lastly, if any unauthorized person tries to pick up the robot to steel it, it automatically send an SMS to the nearest police station with a picture of that thief.

CHAPTER 12
(RESULTS)

12.1 REULTS OF DATASET

In the field of computer vision, convolution neural networks excel at image classification, which consists of categorising images, given a set of classes and having the network determine the strongest class present in the image. The first stage is image acquisition stage. It catches image from camera with the goal that it can be passed for handling and recognition of picture. After picture has been saved, different strategies for handling can be connected to the picture to perform a wide range of vision undertakings. After analyzing, the image is processed and detected. The system is trained using Tensor flow framework. By relying on large datasets, the framework can recognize the picture and plan significant labels and classifications. The trained data is used to classify the waste into Plastic, Paper, Glass and Metal. The video taken by the Pi camera will be processed by the Raspberry Pi. For each frame taken, raspberry pi will detect any garbage present in the frame and send signals to the robot. The edge detection algorithm is used for the differentiation of the scattered edges and compact and collinear edges of the garbage.

12.2 RESULTS OF ROBOT VEHICLE

The robot vehicle is considered to be the main part of the robot. It has customized metal body of Length: 30 cm, Width: 26 cm, Height: 13 cm and Wheel Diameter: 11cm. It is 2 cm above the ground level. All the important devices like Raspberry Pi, Arduino Mega, etc. has been kept inside this vehicle's body. The body contains an extended part in the front to attach the robot arm which will collect the garbage. It has a torque power of 10kg per square cm. We have tested with 15kg weight above the robot and it worked absolutely fine. So, we can say the motors are much more powerful than it was scripted by the company.

12.3 RESULTS OF ROBOT ARM

We have run some experiments regarding the grabbing part of the robot arm. The most available servo motor in Bangladesh is MG996R / MG995. Using this servo, it is quite difficult to grab the object from the ground using 6DOF arm. Though we bought 6 DOF arm but we modified the structure of the arm so that it can grab the objects so frequently. So, our custom built robot arm can pick up garbage of a weight limit of 1kg. We ran some tests earlier on 6DOF and we saw that Shoulder servo falls down with the excessive weight as it's torque is not compatible with that much weight. So, we modified and used two servos together in the shoulder with the help of a customized metallic plate. After modification, our robot arm can grab the garbage and lift it properly. As our robotic hand is compatible with XYZ (3D) rotation so it is enough for our project.

CHAPTER 13 (CONCLUSION)

One of the most adverse impacts of poor waste management, especially municipal waste, is the incidence and prevalence of diseases such as malaria and respiratory problems, as well as other illnesses through the contamination of ground water. Biomedical wastes pose great danger in Bangladesh too as a report estimated that 20% of the biomedical waste is "highly infectious" and is a hazard since it is often disposed of into the sewage system or drains. With regards to the living standards, solid waste leads to blockage in the drainage system which leads to flooding in the streets. Consequently, mosquitoes and bad odour are among the negative impacts resulted. The main objective of this project is to recognize and categorize the waste autonomously, which require minimal human intervention. The robot will collect waste categorizing them as Plastic, Paper, Glass and Metal and put them in a container. Our robot system isolates waste automatically utilizing sensors, motors and detection software. The proposed system concentrates on identification, classification and segregation of waste. As the system works independently, there is no need of human mediation to control or to do any assignment. Utilizing Raspberry Pi and USB camera, the characterization result of images taken will automatically categorize the garbage as Plastic, Paper, Glass and Metal and send signals to the robot to collect the garbage. Once our modified robot gets the signal it will move towards the garbage and collect it using the modified robot arm. Finally the garbage will be dumped in separate portions of a basket based on the type of garbage collected.

Our robot is different from other garbage collector robots as it can run in any unknown location. Sonar sensors allow it to avoid obstacles. Moreover, we used SSD Mobile Lite or detection of garbage which has lower processing power but great detection level. As a result, it will not heat up the processor of Raspberry Pi allowing it to run smoothly. In all, our robot is different from others because it has a high power motors which allows it to move over almost any medium sized stones or rocks. It works in both autonomous and human control modes. It has a custom built obstacle avoidance program which allows it to move in any unknown location. It has a GPS so that if there is any attempt of theft immediate actions can be taken. As we used image processing, the garbage is categorized into Paper, Plastic, Metal and Glass. It has a custom built robot arm that can carry garbage up to 1kg and place it in different sections of the basket attached behind it. As a result, there is no need of any categorization by a human after the garbage has been collected. Lastly, the main objective of this project is to recognize and categorize the waste autonomously, which require minimal human intervention. The robot will collect waste categorizing them as Plastic, Paper, Glass and Metal and put them in a container creating a cleaner and safer environment for humans to live.

CHAPTER 14
(APPENDIX)

14.1 BIBLIOGRAPHY

[1] Nurlansa, Osiany, Dewi Anisa Istiqomah, and Mahendra Astu Sanggha Pawitra. ”AGATOR (automatic garbage collector) as automatic garbage collector robot model.” International Journal of Future Computer and Communication 3.5 (2014): 367.

[2] Wattanasophon, S., and Sarinee Ouitrakul. ”Garbage collection roboton the beach using wireless communications.” Int. Proc. Chem. Biol. Environ. Eng 66 (2014): 92-96.

[3] Rama Prabha, D., Mahindru, S., Srivastava, A., Nilalohita, P. AUTONOMOUS GARBAGE COLLECTOR–ROBODUMPSTER

[4] Ferri, G., Manzi, A., Salvini, P., Mazzolai, B., Laschi, C., Dario, P. (2011, May). DustCart, an autonomous robot for door-to-door garbage collection: From DustBot project to the experimentation in the small town of Peccioli. In 2011 IEEE International Conference on Robotics and Automation (pp. 655-660). IEEE.

[5] Available: https://www.kaggle.com/asdasdasasdas/garbage-classification

[6] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham. Jha, A., Singh, A., Kerketta, R., Prasad, D., Neelam, K., Nath, V. (2019). Development of Autonomous Garbage Collector Robot. In Proceedings of the Third International Conference on Microelectronics, Computing and Communication Systems (pp. 567-576). Springer, Singapore

[7] Simoes, A. S., Carrion, R., Martins, A. C., Costa, R. B., Schvarcz, A. F., Buzo, V. R., ... Ferrari, B. A. (2006, October). Autonomous mobile robots designing for the medical trash collector task. In 2006 IEEE 3rd Latin American Robotics Symposium (pp. 234-239). IEEE.

[8] Parashar, S., Tomar, P. Waste Management by a Robot-A Smart and Autonomous.

[9] Hossain, S., Debnath, B., Anika, A., Junaed-Al-Hossain, M., Biswas, S., Shahnaz, C. (2019, October). Autonomous Trash Collector Based on Object Detection Using Deep Neural Network. In TENCON 2019-2019 IEEE Region 10 Conference (TENCON) (pp. 1406-1410). IEEE.

[10] Nolfi, S. (1997). Evolving non-trivial behaviors on real robots: A garbage collecting robot. Robotics and Autonomous Systems, 22(3-4), 187-198.

[11] Klotzbücher, M., Soetens, P., & Bruyninckx, H. (2010, November). Orocos rtt-lua: an execution environment for building real-time robotic domain specific languages. In International Workshop on Dynamic languages for RObotic and Sensors (Vol. 8).

[12] Brega, R., Tomatis, N., & Arras, K. O. (2000, October). The need for autonomy and real-time in mobile robotics: a case study of XO/2 and Pygmalion. In Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113) (Vol. 2, pp. 1422-1427). IEEE.

[13] Ziemke, T., Carlsson, J., & Bodén, M. (1999, January). An experimental comparison of weight evolution in neural control architectures for a’garbage-collecting’khepera robot. In Proc. of the First Int. Khepera Workshop (pp. 31-40).

[14] Lux, A., & Zoppis, B. (1997). An experimental multi-language environment for the development of intelligent robot systems. In 5th International Symposium on Intelligent Robotic Systems, SIRS (Vol. 97, pp. 169-174).

[15] Maude J, Ramnarayan P, Tomlinson A. The Impact of a Web-based Diagnosis Checklist System on Specialist Referrals from Primary Care: Results of a Survey of General

Practitioners. 2010; see: http://www.isabelhealthcare.com/home/peer/ new1 (last checked 13 February 2012)

[16] Amy L, Borowitz S, Brown P, Mendelsohn M, Lyman J, Eds. The impact of a web-based diagnosis reminder system on errors of diagnosis. American Medical Informatics Association Symposium 2006

[17] Garg AX, Adhikari NKJ, McDonald H, et al. Effects of computerized clinical decision support systems on practitioner performance and patient outcomes. JAMA 2005;293:1223–38

[18] "Path lab Online Laboratory Information Management Software - Laboratory Information system, LIMS software, Online Diagnostic software India", Epathlab.com, 2016. [Online].Available:http://www.epathlab.com/epathlab-lims-how-it-works.html. [Accessed: 20- Nov- 2016].

[19] Available: http://www.atlas.d-waste.com/

[20] Available: http://wasteconcern.org/Publication/Waste%20Survey_05.pdf

[21] Available: http://www.apo-tokyo.org/publications/files/ind-22-swm.pdf

[22] Available: https://www.daily-sun.com/printversion/details/346071/2018/10/28/Waste-Management-in-Bangladesh

[23] Available: http://www.ijfcc.org/papers/329-CS3004.pdf

[24] Available: http://www.citylab.com/tech/2016/03/the-robot-garbage-collectors-are-coming/471429/

[25] Available: http://www.ipcbee.com/vol66/019-IEEA2014-A044.pdf

[26] Available: https://github.com/tensorflow/models/tree/master/research/object_detection

[27] Available: https://github.com/topics/object-detection

[28] Available: https://github.com/bourdakos1/Custom-Object-Detection

[29] Available: https://www.tensorflow.org/lite/models/object_detection/overview

[30] Available: https://tensorflow-object-detection-api- Available: tutorial.readthedocs.io/en/latest/training.html

[31] Available: https://github.com/EdjeElectronics/TensorFlow-Object-Detection-on-the-Raspberry-Pi

[32] Available: https://www.pyimagesearch.com/2017/10/16/raspberry-pi-deep-learning-object-detection-with-opencv/

[33] Available: https://www.kaggle.com/andreasantoro/split-garbage-dataset?fbclid=IwAR1iBR17E8RpiM-1K34tDuOKS1AFSfRZ-xA5DnWRqzSNft9NC51wmoVpd_M

[34] Available: https://medium.com/@jonathan_hui/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06

[35] Available: https://medium.com/@smallfishbigsea/understand-ssd-and-implement-your-own-caa3232cd6ad

[36] Available: http://article.sapub.org/10.5923.j.ijit.20170602.06.html

14.2 CODE FOR DETECTION OF GARBAGE

```
# Import packages

import os

import cv2

import numpy as np

import tensorflow as tf

import sys


# This is needed since the notebook is stored in the object_detection folder.

sys.path.append("..")


# Import utilites

from utils import label_map_util

from utils import visualization_utils as vis_util


# Name of the directory containing the object detection module we're using

MODEL_NAME = 'garbage_model'


# Grab path to current working directory

CWD_PATH = os.getcwd()


# Path to frozen detection graph .pb file, which contains the model that is used

# for object detection.

PATH_TO_CKPT = os.path.join(CWD_PATH,MODEL_NAME,'frozen_inference_graph.pb')


# Path to label map file
```

```
PATH_TO_LABELS = os.path.join(CWD_PATH,'training','labelmap.pbtxt')


# Number of classes the object detector can identify
NUM_CLASSES = 4


## Load the label map.
# Label maps map indices to category names, so that when our convolution
# network predicts `5`, we know that this corresponds to `king`.
# Here we use internal utility functions, but anything that returns a
# dictionary mapping integers to appropriate string labels would be fine
label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map,
max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)


# Load the Tensorflow model into memory.
detection_graph = tf.Graph()
with detection_graph.as_default():
    od_graph_def = tf.GraphDef()
    with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
        serialized_graph = fid.read()
        od_graph_def.ParseFromString(serialized_graph)
        tf.import_graph_def(od_graph_def, name='')


    sess = tf.Session(graph=detection_graph)
```

```
# Define input and output tensors (i.e. data) for the object detection classifier

# Input tensor is the image
image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')

# Output tensors are the detection boxes, scores, and classes
# Each box represents a part of the image where a particular object was detected
detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')
# Each score represents level of confidence for each of the objects.
# The score is shown on the result image, together with the class label.
detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')
detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')

# Number of objects detected
num_detections = detection_graph.get_tensor_by_name('num_detections:0')
# Initialize webcam
video = cv2.VideoCapture(0)
ret = video.set(3,1280)
ret = video.set(4,720)

while(True):

    # Acquire frame and expand frame dimensions to have shape: [1, None, None, 3]
    # i.e. a single-column array, where each item in the column has the pixel RGB value
    ret, frame = video.read()
```

```
    frame_expanded = np.expand_dims(frame, axis=0)

    # Perform the actual detection by running the model with the image as input
    (boxes, scores, classes, num) = sess.run(
        [detection_boxes, detection_scores, detection_classes, num_detections],
        feed_dict={image_tensor: frame_expanded})
    # Draw the results of the detection (aka 'visulaize the results')
    vis_util.visualize_boxes_and_labels_on_image_array(
        frame,
        np.squeeze(boxes),
        np.squeeze(classes).astype(np.int32),
        np.squeeze(scores),
        category_index,
        use_normalized_coordinates=True,
        line_thickness=8,
        min_score_thresh=0.60)

    # All the results have been drawn on the frame, so it's time to display it.
    cv2.imshow('Object detector', frame)
    # Press 'q' to quit
    if cv2.waitKey(1) == ord('q'):
        break
# Clean up
video.release()
cv2.destroyAllWindows()
```

14.3 CODE FOR AUTONOMOUS GARBAGE COLLECCTION

```
/* Garbage Detection and Collection Robot using Image Processing
  Device used: Arduino Mega 2560
  This code represents the mechanism of the hardwares of the robot
  Code includes two mode: Human controlled
                           Autonomous

  Functions:    Robot Movement, Robotic Hand Functions, Basket Rotation,
             Error Detections, Alert System, Security

  Coded By:     Moin Shahriyar
             B.Sc. in CSE
             North South University
*/
#include <VarSpeedServo.h>
#include <elapsedMillis.h>




//from below to upwards (1-3)
VarSpeedServo servo1;
VarSpeedServo servo2;
VarSpeedServo servo3;
VarSpeedServo servo4;
VarSpeedServo servo5;
VarSpeedServo servo6;
```

```
VarSpeedServo basketServo;

VarSpeedServo camServo;



//vcc pin

const int vcc_1 = 13;


//read pins

const int auto_mode = 23;


const int forward = 25;

const int backward = 27;

const int left = 29;

const int right = 31;


const int grab = 33;


const int glass = 40;

const int metal = 42;

const int paper = 44;

const int plastic = 46;


//write pins

const int relay_1 = 22;
```

```
const int relay_2 = 24;
const int relay_3 = 26;
const int relay_4 = 28;

//pwm pins
const int servo_1 = 7;
const int servo_2 = 6;
const int servo_3 = 5;
const int servo_4 = 4;
const int servo_5 = 3;
const int servo_6 = 2;

const int basket_servo = 8;
const int cam_servo = 9;


//sonar for grabbing
const int grab_trigPin = 34;
const int grab_echoPin = 35;

//sonar for obstacle avoidance
//right sonar
const int obs_trigPin_1 = 36;
const int obs_echoPin_1 = 37;

//left sonar
```

```
const int obs_trigPin_2 = 38;

const int obs_echoPin_2 = 39;


//time diff for turning at an angle

elapsedMillis timeDiff = 0;


//random movement time

const int moveTime = 1500; //1500 millisec = 1.5 sec

/*

  it takes ~8.5 sec to cross 435 cm

  so ~51cm per sec.

  ~102cm for 2 sec.

*/


//rotation time for moving at an angle

const int rotateTime = 415;

/*

  this robot takes 32~33 sec for 10 rotation of 360 degree

  so 1 rotation of 360 degree needs 3.2~3.3 sec

  so 45 degree of rotation needs (3.3/8)=0.4125 sec ~ 412.5 millisec

  Note: This was measured with fully charged battery.

  If voltage decreases time for rotation will increase.

*/



void setup() {
```

```
Serial.begin(9600);

//vcc pin
pinMode(vcc_1, OUTPUT);
digitalWrite(vcc_1, HIGH);

//servo_setup
//default_hand_code
default_hand();



//cam servo
camServo.attach(cam_servo);
camServo.write(45, 50, true);

//basket servo
basketServo.attach(basket_servo);
basketServo.write(90, 50, true);

//sonar for grabbing
pinMode(grab_trigPin, OUTPUT);

//sonar for obstacle avoidance
pinMode(obs_trigPin_1, OUTPUT);
pinMode(obs_trigPin_2, OUTPUT);
```

```
  //motor_setup
  pinMode(relay_1, OUTPUT);
  pinMode(relay_2, OUTPUT);
  pinMode(relay_3, OUTPUT);
  pinMode(relay_4, OUTPUT);

  //all relays set to high to deactivate
  reset_relay();



  Serial.println("Setup Completed");

}

void loop() {

  if (digitalRead(auto_mode) == HIGH) {
    auto_bot();
  }
  else {
    digitalWrite(auto_mode, LOW);
    human_control();
  }

}
```

```
void auto_bot() {
  Serial.println("start");
  reset_relay();

  //auto drive code goes here
  if (check_default_hand() == 0) {
    default_hand();
  }

  timeDiff = 0;
  Serial.print("timeDiff");
  Serial.println(timeDiff);

  //max distance of object
  int dist = 0;

  //random movement [idea & code from "FARJANA ALAM"]
  int rand_array[] = {forward, left, right};
  int movement = 0;


  //intiate pi cam servo and get angle
  //int camAngle = cam_angle();
  timeDiff = 0;
  //Serial.println("camAngle");
  //Serial.println(camAngle);
```

```
while (check_distance(grab_trigPin, grab_echoPin) > 10
       && check_distance(grab_trigPin, grab_echoPin) <= 50) {
  if (digitalRead(auto_mode) != HIGH) {
    human_control();
  }
  else {
    int dist = check_distance(grab_trigPin, grab_echoPin);
    if (dist >= 11 && dist <= 13) {
      reset_relay();
      if (check_distance(obs_trigPin_1, obs_echoPin_1) > 30 &&
          check_distance(obs_trigPin_2, obs_echoPin_2) > 30) {
        timeDiff = 0;
        while (timeDiff <= 10000) {
          check_type();
        }
        Serial.println("Object found");
        grab_object();
        return 0;
      }
      else {
        return 0;
      }


    }
    else {
      if (check_distance(obs_trigPin_1, obs_echoPin_1) > 30 &&
```

```
        check_distance(obs_trigPin_2, obs_echoPin_2) > 30) {
      move_forward();
    }
    else {
      reset_relay();
      return 0;
    }
    Serial.println("moving forward to grab");
  }
}

}



Serial.println("randomMove");

movement = rand_array[random(0, 3)]; //excluding max value
if (movement == forward) {
  timeDiff = 0;
  while (timeDiff < moveTime) {
    if (digitalRead(auto_mode) != HIGH) {
      human_control();
    }
    else {
      if (check_distance(obs_trigPin_1, obs_echoPin_1) > 30 &&
          check_distance(obs_trigPin_2, obs_echoPin_2) > 30) {
```

```
        timeDiff = 0;
        while (timeDiff < moveTime) {
          move_forward();
        }
        reset_relay();
        return 0;
      }
      else {
        reset_relay();
        return 0;
      }
    }
  }
  reset_relay();
  delay(5000);
  timeDiff = 0;
  return 0;
}
else if (movement == left) {
  timeDiff = 0;
  while (timeDiff < rotateTime) {
    if (check_distance(obs_trigPin_2, obs_echoPin_2) > 30) {
      move_left();
    }
    else {
      reset_relay();
```

```
      return 0;
    }
  }
  reset_relay();
  delay(5000);
  timeDiff = 0;
  while (timeDiff < moveTime) {
    if (digitalRead(auto_mode) != HIGH) {
      human_control();
    }
    else {
      if (check_distance(obs_trigPin_1, obs_echoPin_1) > 30 &&
          check_distance(obs_trigPin_2, obs_echoPin_2) > 30) {
        move_forward();
      }
      else {
        reset_relay();
        return 0;
      }
    }
  }
  reset_relay();
  delay(5000);
  timeDiff = 0;
  return 0;
}
```

```
else if (movement == right) {
  timeDiff = 0;
  while (timeDiff < rotateTime) {
    if (check_distance(obs_trigPin_1, obs_echoPin_1) > 30) {
      move_right();
    }
    else {
      reset_relay();
      return 0;
    }
  }
  reset_relay();
  delay(5000);
  timeDiff = 0;
  while (timeDiff < moveTime) {
    if (digitalRead(auto_mode) != HIGH) {
      human_control();
    }
    else {
      if (check_distance(obs_trigPin_1, obs_echoPin_1) > 30 &&
          check_distance(obs_trigPin_2, obs_echoPin_2) > 30) {
        move_forward();
      }
      else {
        reset_relay();
        return 0;
```

```
      }
    }
  }
  reset_relay();
  delay(5000);
  return 0;
 }



 Serial.println("auto_bot");
 return 0;
}



void human_control() {
 if (digitalRead(forward) == HIGH &&
     check_distance(obs_trigPin_1, obs_echoPin_1) > 30 &&
     check_distance(obs_trigPin_2, obs_echoPin_2) > 30) {
  move_forward();
 }
 else if (digitalRead(backward) == HIGH) {
  move_backward();
 }
 else if (digitalRead(left) == HIGH &&
```

```
        check_distance(obs_trigPin_2, obs_echoPin_2) > 30) {
    move_left();
  }
  else if (digitalRead(right) == HIGH &&
        check_distance(obs_trigPin_1, obs_echoPin_1) > 30) {
    move_right();
  }
  else {
    reset_relay();
  }
  if (digitalRead(grab) == HIGH) {
    //int dist = check_distance(grab_trigPin, grab_echoPin);
    //if (dist >= 11 && dist <= 13) {
    reset_relay();
    timeDiff = 0;
    while (timeDiff <= 4000) {
      check_type();
    }
    grab_object();
    //} else {
    // Serial.println("Insufficient Distance");
    //}

  }

}
```

```
void move_forward() {
  /*int dist_1 = check_distance(obs_trigPin_1, obs_echoPin_1);
    int dist_2 = check_distance(obs_trigPin_2, obs_echoPin_2);
    if (dist_1 > 30 && dist_2 > 30) {*/
  digitalWrite(relay_1, HIGH);
  digitalWrite(relay_2, LOW);
  digitalWrite(relay_3, HIGH);
  digitalWrite(relay_4, LOW);
  Serial.println("forward");
  //}

}
void move_backward() {
  digitalWrite(relay_1, LOW);
  digitalWrite(relay_2, HIGH);
  digitalWrite(relay_3, LOW);
  digitalWrite(relay_4, HIGH);
  Serial.println("backward");
}
void move_left() {
  /*int dist_2 = check_distance(obs_trigPin_2, obs_echoPin_2);
    if (dist_2 > 30) {*/
  digitalWrite(relay_1, HIGH);
  digitalWrite(relay_2, LOW);
  digitalWrite(relay_3, LOW);
```

```
  digitalWrite(relay_4, HIGH);
  Serial.println("left");
  //}
}
void move_right() {
  /*int dist_1 = check_distance(obs_trigPin_1, obs_echoPin_1);
    if (dist_1 > 30) {*/
  digitalWrite(relay_1, LOW);
  digitalWrite(relay_2, HIGH);
  digitalWrite(relay_3, HIGH);
  digitalWrite(relay_4, LOW);
  Serial.println("right");
  //}
}
void grab_object() {
  reset_relay();      //bot should stop moving
  //robot_hand function goes here
  Serial.println("grabbing");

  //grab
  servo1.write(100, 50, true);
  delay(1000);
  servo2.write(80, 20, true);
  delay(1000);
  servo6.attach(servo_6);
  servo6.write(45, 50, true);
```

```
delay(1000);
servo6.detach();
servo5.write(0, 50, true);
delay(1000);
servo3.write(165, 30, true);    //previous: 170
delay(1000);

servo4.write(180, 50, true);
delay(1000);


servo6.attach(servo_6);
servo6.write(90, 50, true);
delay(1000);
servo4.write(90, 50, true);
delay(1000);
servo3.write(90, 30, true);
delay(1000);
servo2.write(90, 20, true);
delay(1000);

Serial.println("Grabbing Complete");

//store

servo3.write(90, 30, true);
```

```
  delay(1000);
  servo4.write(70, 40, true);
  delay(1000);
  servo5.write(180, 50, true);
  delay(1000);
  servo6.write(45, 50, true);
  delay(1000);
  servo6.write(90, 50, true);
  delay(1000);
  servo6.detach();

  Serial.println("Storing Complete");

  default_hand();

}

void default_hand() {
  //should be upside down (6-1)
  Serial.println("Default hand");
  servo6.attach(servo_6);
  servo6.write(90, 50, true);
  delay(1000);
  servo6.detach();
  servo5.attach(servo_5);
  servo5.write(90, 50, true);
```

```
  delay(1000);
  servo4.attach(servo_4);
  servo4.write(90, 50, true);
  delay(1000);
  servo3.attach(servo_3);
  servo3.write(90, 50, true);
  delay(1000);
  servo2.attach(servo_2);
  servo2.write(90, 50, true);
  delay(1000);
  servo1.attach(servo_1);
  servo1.write(100, 50, true);

  delay(6000);
}

int check_distance(int trigPin, int echoPin) {
  long duration;
  int distance;
  //make sure trig pin is clear
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  //start measuring distance using duration
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
```

```
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;

  return distance;

}

//auto_bot function
int cam_angle() {
  int camAngle = -1;          //nothing is found

  camServo.write(0, 90, true);
  delay(10000);
  if (check_type() == 1) {
    camAngle = 0;
    camServo.write(45, 90, true);  //reset camServo
    return camAngle;          //at 45 degree left garbage is found
  }
  camServo.write(45, 90, true);
  delay(10000);               //fix delay later if needed
  if (check_type() == 1) {
    camAngle = 45;
    camServo.write(45, 90, true);  //reset camServo
    return camAngle;          //at straight garbage is found
  }
```

```
  camServo.write(90, 90, true);
  delay(10000);
  if (check_type() == 1) {
    camAngle = 90;
    camServo.write(45, 90, true);  //reset camServo
    return camAngle;          //at 45 degree right garbage is found
  }
  camServo.write(45, 0, true); //reset camServo
  return camAngle;            //nothing is found
}

int check_type() {
  if (digitalRead(glass) == HIGH) {
    //rotate basket servo


    return 1;
  }
  else if (digitalRead(metal) == HIGH) {
    //rotate basket servo
    basketServo.write(0, 90, true);
    return 1;
  }
  else if (digitalRead(paper) == HIGH) {
    //rotate basket servo
    basketServo.write(90, 90, true);
    return 1;
```

```
  }
  else if (digitalRead(plastic) == HIGH) {
    //rotate basket servo
    basketServo.write(180, 90, true);
    return 1;
  }
  else {
    return 0;
  }
}
void reset_relay() {
  digitalWrite(relay_1, HIGH);
  digitalWrite(relay_2, HIGH);
  digitalWrite(relay_3, HIGH);
  digitalWrite(relay_4, HIGH);
  Serial.println("All Relays are reset");
}
int check_default_hand() {
  if (servo2.read() == 90 && servo3.read() == 90 &&
      servo4.read() == 90 && servo5.read() == 90 && servo6.read() == 90) {
    return 1;
  } else {
    return 0;
  }
  return 0;
}
```