INTELLIGENT WASTE CLASSIFICATION FOR A SUSTAINABLE FUTURE

A PROJECT REPORT

Submitted by

SELVAKUMAR P	**(211419105132)**
SASIDARAN S	**(211419105130)**
VIGNESH K	**(211419105163)**
VIGNESH V	**(211419105165)**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

PANIMALAR ENIGNEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2023

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"INTELLIGENT WASTE CLASSIFICATION FOR A SUSTAINABLE FUTURE"** is the work of **"SELVAKUMAR P (211419105132), SASIDARAN S (211410105130), VIGNESH K (211419105163), VIGNESH V (211419105165) "** who carried out the project under my supervision.

Signature	Signature
Dr. S. SELVI, M.E, Ph.D.	**Mr. N. KARTHIKEYAN**
PROFESSOR	SUPERVISOR
Head of the Department	Assistant Professor
Department of Electrical and	Department of Electrical and
Electronics Engineering,	Electronics Engineering,
Panimalar Engineering College,	Panimalar Engineering College,
Chennai -123	Chennai – 123

ACKNOWLEDGEMENT

Our sincere thanks to our Honourable Founder and Chairman, **Dr. JEPPIAAR, M.A., B.L., Ph.D.,** for his sincere endeavour in educating us in his premier institution.

We would like to express our deep gratitude to our beloved Secretary and correspondent, **Dr. P. CHINNADURAI, M.A., M.Phil.,** Ph., for his enthusiastic motivation which inspired us a lot in completing this project and our sincere thanks to our Directors **Mrs. C. VIJAYA RAJESWARI, Dr. C. SHAKTHI KUMAR, M.E., Ph.D., AND Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for providing us with necessary facilities for the completion of this project.

We would like to express thanks to our Principal, **Dr. K. MANI, M.E., Ph.D.,** for extended his guidance and cooperation.

We would also like to thank our Head of the Department, **Dr. S. SELVI, M.E., Ph.D., Professor and Head, Department of Electrical and Electronics Engineering** for her encouragement.

Personally, we thank our guide **Mr. N. KARTHIKEYAN, Assistant Professor, in Department of Electrical and Electronics Engineering** for the persistent motivation and support for this project, who at all times was the mentor of germination of this project from a small idea.

We express our sincere thanks to the project coordinators **Dr. S. DEEPA & Dr. N. MANOJ KUMAR, M.E., Ph.D., in Department of Electrical and Electronics Engineering** for the valuable Suggestions from time to time at every stage of our project.

Finally, we would like to take this opportunity to thank our family members, faculty and non-teaching staff members of our department, friends, well-wishers who have helped us for the successful completion of our project.

ABSTRACT

Cities are constantly working to improve their garbage handling facilities. But garbage segregation is still one of the biggest challenges that needs to be tackled. Existing systems, such as robots, can be cost prohibitive. The task of garbage segregation can be divided into garbage separation and garbage sorting. In this project, we propose a low-cost way to sort garbage using a light weight classification model called TensorFlow Lite.

This machine learning model is embedded in a custom-made application for Android phone. This eliminates the need for external camera and image processing hardware. To demonstrate the working of this application, a prototype to sort garbage has been built. The phone is set on top of a conveyor. A cheap microcontroller called ESP8266 is used to monitor incoming garbage, control the conveyor, communicate with the phone and sort the garbage. By using a smart phone and an inexpensive microcontroller, the proposed system provides a cost-effective, automated approach that can be implemented at a community level and help to provide a sustainable future.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATION

ESP	Espressif
CNN	Convolutional Neural Network
VGG	Visual Geometry Group
AI	Artificial Intelligence
GPU	Graphics Processing unit
DSP	Digital Signal Processor
NPU	Neural Processing Unit
IOT	Internet of Things
IDE	Integrated Development Environment
MCU	Micro-Controller unit
DC	Direct Current
APK	Application
TCP	Transmission Control Protocol
IP	Internet Protocol
LED	Light-emitting diode
DIY	Do it yourself
ML	Machine Learning
TT	Textile Textile
HTTP	Hypertext Transfer Protocol

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Waste management is a critical environmental issue that requires urgent attention. Improper waste disposal practices can lead to negative impacts on public health, the environment, and the economy. One of the biggest challenges in waste management is waste segregation, the process of separating different types of waste for proper treatment and disposal. However, waste segregation can be a labour-intensive and time-consuming process that can lead to human error and improper segregation.

To address the challenges of waste segregation, this project proposes an intelligent waste classification system that uses machine learning to automatically sort different types of waste. The system is designed to be low-cost and scalable, making it suitable for use in communities and households. The proposed solution uses a lightweight classification model called TensorFlow Lite, which is embedded in a custom-made application for Android phone. This eliminates the need for external camera and image processing hardware, making the system more affordable and accessible.

The proposed system involves a conveyor system where objects are placed and transported for sorting. When an object is detected at a certain point, the conveyor stops, and the object is scanned using a TensorFlow model. The output is then sent to an ESP8266, which controls the conveyor and tilts the plate to sort the object. This process is repeated for each object placed on the conveyor, enabling efficient and automated sorting.

The goal of the project is to provide an efficient and affordable solution for waste management that reduces the environmental impact of waste disposal and promotes sustainable practices. By automating the waste segregation process, the system can reduce the amount of time and labour required for waste management and improve the accuracy and efficiency of waste segregation. The proposed solution has the potential to promote sustainable waste management practices by making it easier for individuals and communities to sort waste and dispose of it in an environmentally responsible manner.

In this report, we provide a detailed description of the proposed system, including the hardware and software components and the design and implementation of the system. We also present the results of our experiments and discuss the potential applications and limitations of the proposed system. The report concludes with a summary of the main findings and recommendations for future research and development in the field of waste management.

1.2 BACKGROUND AND MOTIVATION

Waste management is one of the most pressing environmental challenges of our time. Rapid urbanization and population growth have led to a significant increase in the amount of waste generated, and waste management systems in many parts of the world are struggling to keep up with the demand. Inefficient and inadequate waste management practices can have significant negative impacts on public health, the environment, and the economy. For example, improper disposal of waste can lead to air and water pollution, increased greenhouse gas emissions, and the spread of disease.

One of the biggest challenges in waste management is waste segregation. The separation of different types of waste is necessary to ensure that each type of waste is treated and disposed of in the most appropriate manner. For example, organic waste can be composted or used for biogas production, while non-biodegradable waste can be recycled or disposed of in a landfill. However, waste segregation can be a labour-intensive and time-consuming process, and human error can lead to improper segregation. The motivation for this project is to address the challenges of waste segregation and promote sustainable waste management practices. The goal of the project is to provide an efficient and affordable solution for waste management that reduces the environmental impact of waste disposal and promotes sustainable practices. The proposed solution is an intelligent waste classification system that uses machine learning to automatically sort different types of waste.

The system is designed to be low-cost and scalable, making it suitable for use in communities and households. By automating the waste segregation process, the system can reduce the amount of time and labour required for waste management and improve the accuracy and efficiency of waste segregation. This can have significant positive impacts on public health, the environment, and the economy.

The proposed system also has the potential to promote sustainable waste management practices by making it easier for individuals and communities to sort waste and dispose of it in an environmentally responsible manner. By reducing the amount of waste that is improperly disposed of, the system can help to reduce the negative impacts of waste on the environment and promote a more sustainable future.

Overall, the background and motivation for this project stem from the urgent need to address the challenges of waste management and promote sustainable practices. The proposed solution has the potential to make a significant contribution to the field of waste management and help to create a more sustainable future for all.

1.3 OBJECTIVE

The objective of this project is to develop an intelligent waste classification system that uses machine learning to automatically sort different types of waste. The proposed system aims to provide an affordable and scalable solution for waste management, promoting sustainable practices and reducing the environmental impact of waste disposal. The block diagram of the system is shown in Fig 1.1.

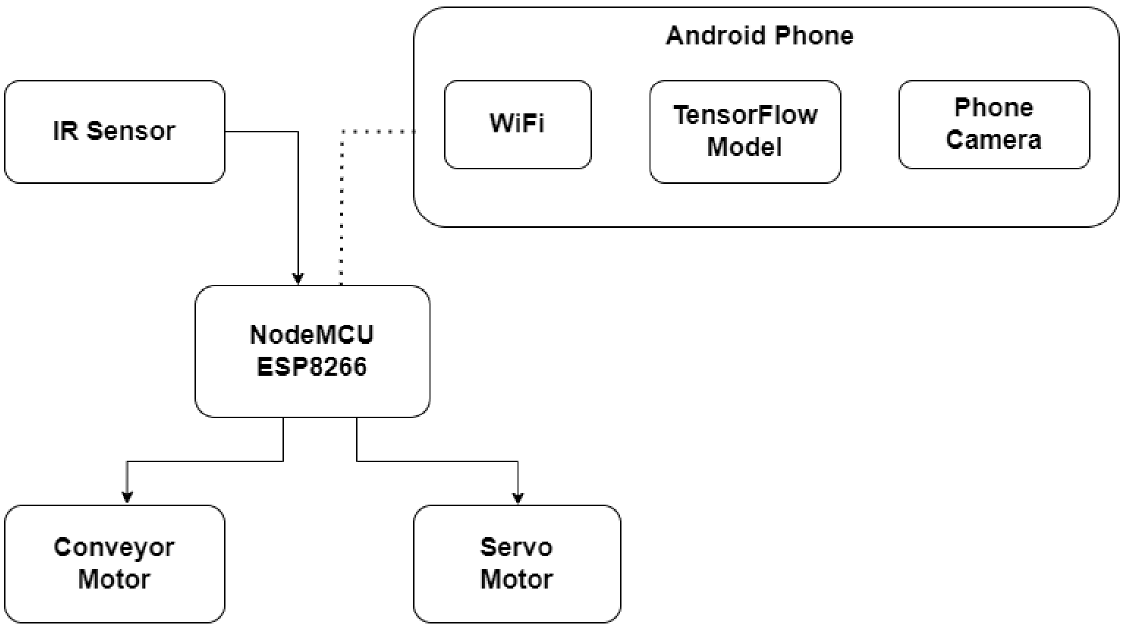


Fig 1.1 System block diagram

Specifically, the project aims to:

- Develop a lightweight machine learning model using TensorFlow Lite that can accurately classify different types of waste.
- Design and implement a low-cost hardware system for waste sorting, using an Android phone and an ESP8266 microcontroller.
- Evaluate the performance of the system in terms of accuracy, speed, and cost-effectiveness.
- Demonstrate the feasibility and potential applications of the proposed system for waste management at a community level.
- Contribute to the development of sustainable waste management practices by making waste segregation more efficient and accessible.

The ultimate goal of this project is to contribute to a sustainable future by promoting responsible waste management practices and reducing the negative environmental impact of waste disposal.

1.4 SCOPE AND LIMITATIONS

1.4.1 Scope

Contribute to the development of sustainable waste management practices by making waste segregation more efficient and accessible. The aim of this project is to contribute to a sustainable future by promoting responsible waste management practices and reducing the negative environmental impact of waste disposal.

The system will be evaluated in terms of its accuracy, speed, and cost-effectiveness, with the aim of demonstrating the feasibility and potential applications of the proposed system for waste management at a community level. The project will be limited to the use of visual and moisture sensors to detect and classify waste types.

The scope of this project is to develop a waste classification system and build a prototype to showcase the effectiveness of the classification by sorting. It should be noted that the project focuses on garbage sorting and not on garbage separation, which involves a different set of equipment and processes. The development of garbage separation equipment is beyond the scope of this project.

1.4.2 LIMITATIONS:

There are several limitations to the proposed system. Firstly, the accuracy of the waste classification system may be affected by the quality and variability of the input data, which could result in misclassifications. Secondly, the proposed system is designed to sort only a limited set of waste types, and may not be suitable for sorting more complex or specialized waste categories.

Thirdly, the hardware system used in this project is limited by the processing power and memory capacity of the Android phone and the ESP8266 microcontroller. This may limit the scale and speed at which the system can operate. Fourthly, the proposed system may not be suitable for all geographical and environmental contexts, and may require modifications and adaptations to suit different locations and waste management systems. It should be also noted that this system can only process one garbage object at a time.

Lastly, the system may require regular maintenance and calibration to ensure accurate and consistent performance. Despite these limitations, the proposed system offers a cost-effective and accessible solution for waste classification and management, with the potential to contribute to a more sustainable future.

CHAPTER 2

LITERATURE REVIEW

2.1 EXISTING SYSTEM

In recent years, deep learning-based approaches have shown great success in image classification tasks. Convolutional Neural Networks (CNNs) are the most popular deep learning architectures used for image classification tasks. CNNs are designed to extract features from images and classify them into different categories. There are several CNN architectures like VGG, ResNet, Inception, MobileNet, etc., that are popularly used for image classification tasks.

VGG (Visual Geometry Group) is a popular CNN architecture that was developed by researchers at the University of Oxford. VGG has multiple layers with small filters that help in extracting more detailed features from the images. The VGG architecture is known for its simplicity and its ability to extract high-level features from the images. VGG has been used in various image classification tasks, including garbage classification. The typical structure of VGG is shown in Fig 2.1.

ResNet (Residual Network) is another popular CNN architecture that was developed by researchers at Microsoft Research. ResNet is known for its ability to train very deep neural networks. ResNet uses a residual block that helps in training deeper neural networks without overfitting. ResNet has shown state-of-the-art performance in various image classification tasks, including garbage classification.

Garbage classification is a challenging task as it involves classifying various types of waste materials, including organic and inorganic wastes. The existing systems for garbage classification often use CNN architectures like VGG and ResNet for image classification tasks. However, these architectures require high processing and memory requirements, making them costly and impractical for low-cost applications.

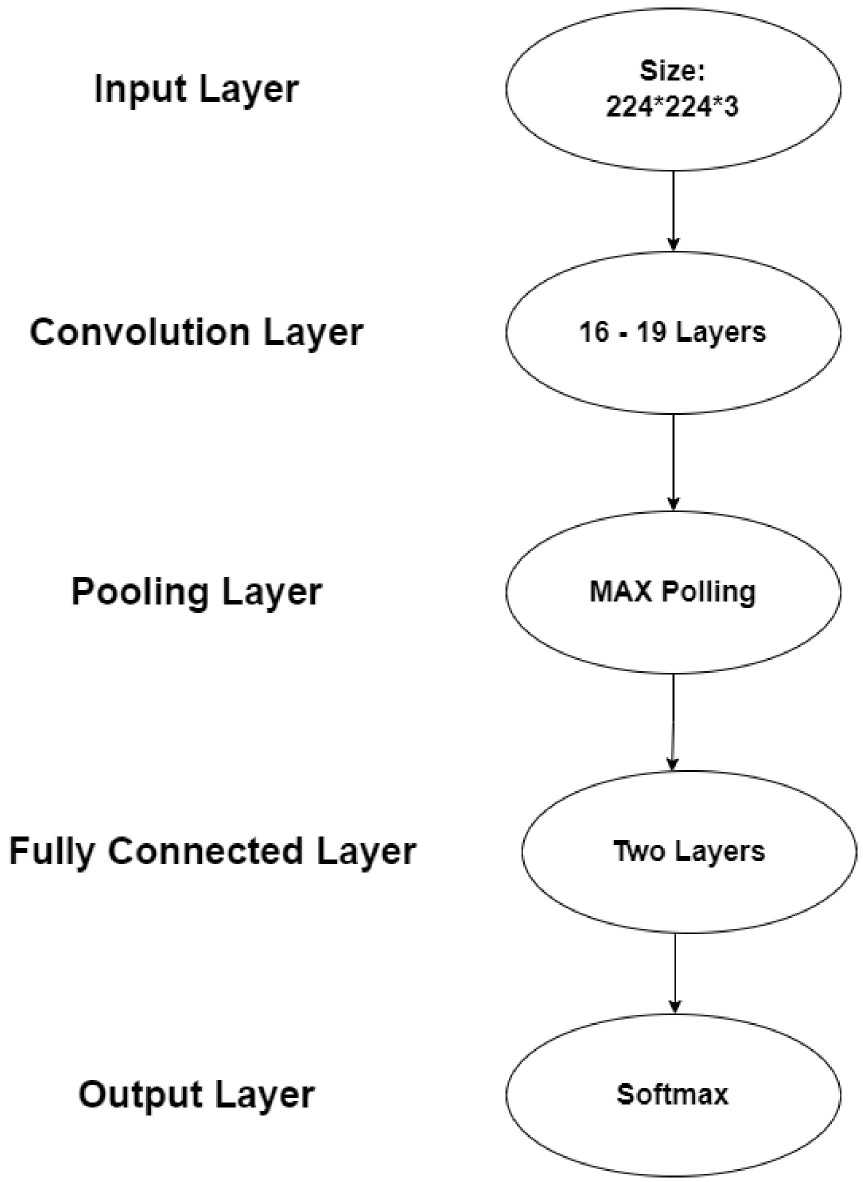


Fig 2.1 Existing System

2.2 DISADVANTAGES OF EXISTING SYSTEM

The existing systems require external hardware like an external image processing system like Raspberry Pi. These systems can be expensive, making them unsuitable for implementation in low-resource settings. Additionally, the high computational requirements of deep learning algorithms like ResNet and VGG make them computationally expensive, making them unsuitable for deployment on low-cost microcontrollers.

2.3 PROPOSED SYSTEM

The proposed system uses a lightweight convolutional neural network architecture called MobileNet, which is specifically designed for mobile and embedded devices. Compared to other popular CNN architectures like VGG and ResNet, MobileNet requires significantly less processing power and memory, making it ideal for resource-constrained devices like smartphones and microcontrollers. The use of MobileNet enables the proposed system to be cost-effective and highly efficient for garbage classification. The typical structure of MobileNet is shown in Fig 2.2.

To demonstrate the capabilities of the proposed system, a prototype has been built that utilizes an Android phone and an ESP8266 microcontroller. The phone is mounted on a stand above a conveyor, which is used to scan the garbage objects. An IR sensor is used to detect the presence of an object, and the captured image is fed into a custom-made Android application containing the MobileNet model. The model analyses the image and outputs whether the detected object is biodegradable or non-biodegradable, and the microcontroller tilts a plate to sort the object accordingly.

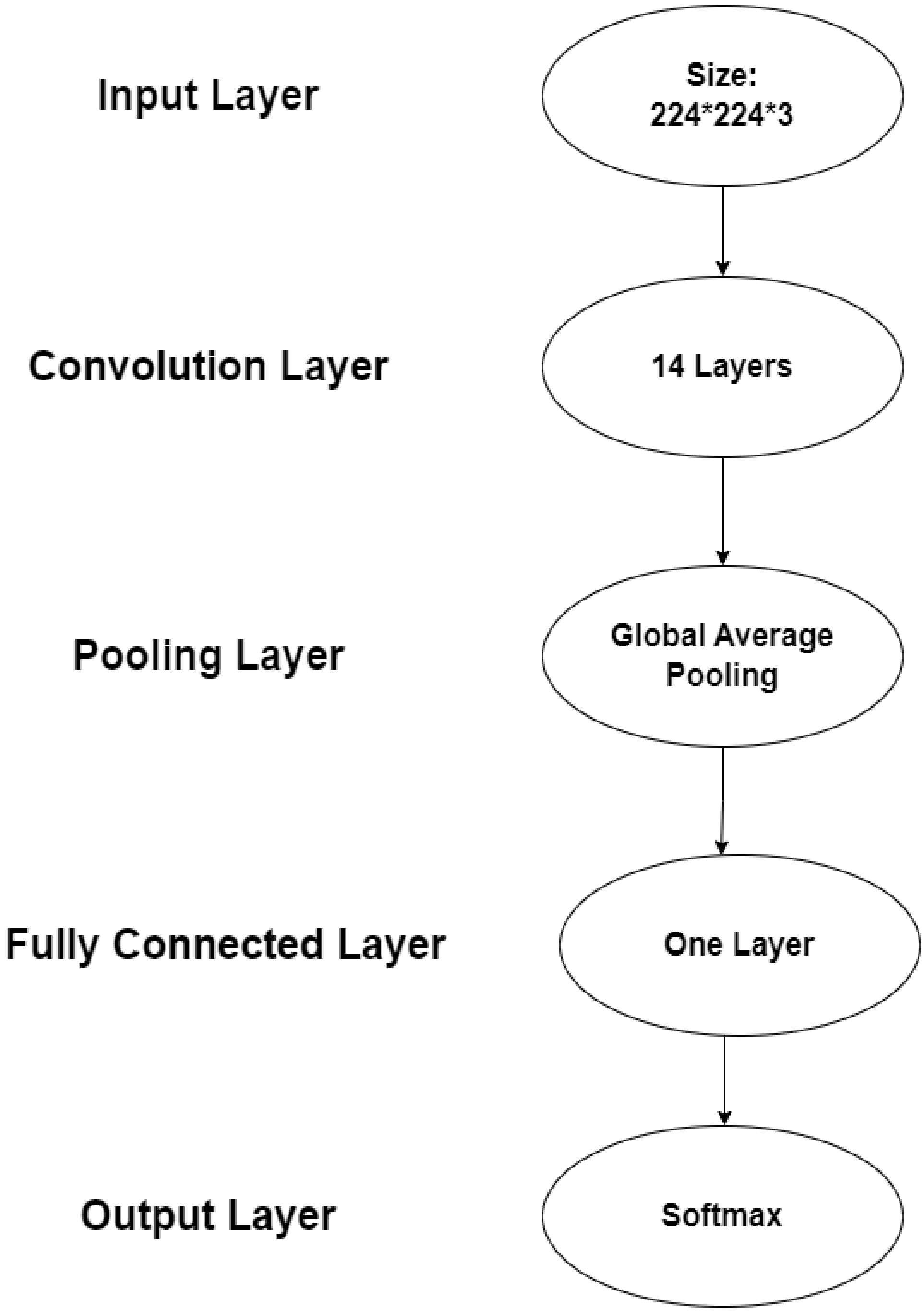


Fig 2.2 Proposed System

2.3.1 Quantized MobileNet

In our project we have used Quantized MobileNet as the Algorithm. Quantized MobileNet is an optimized version of MobileNet, a popular neural network architecture for image classification tasks that was developed by Google. This algorithm uses quantization techniques to reduce the memory and computational requirements of MobileNet, making it an ideal choice for edge devices with limited computing power. The working of Quantized MobileNet is shown in Fig 2.3.

Quantization is a technique that reduces the precision of weights and biases in a neural network from 32 bits to lower bit precision, such as 8 bits. This significantly reduces the memory required to store these parameters, as well as the computational power required to perform calculations on them. In addition, quantization can improve the energy efficiency of the neural network.

In the case of Quantized MobileNet, the weights and biases are quantized to 8 bits, which reduces the memory requirements of the network by a factor of 4. This allows the network to be run on devices with limited memory, such as mobile phones and other edge devices.

Quantized MobileNet also employs a technique called depthwise separable convolution, which is a modification of the standard convolutional operation used in neural networks. This technique breaks down the standard convolution into two smaller operations: depthwise convolution and pointwise convolution. The depthwise convolution applies a single filter to each input channel, while the pointwise convolution applies a 1x1 filter to the output of the depthwise convolution. This reduces the number of parameters in the network, while maintaining its accuracy.

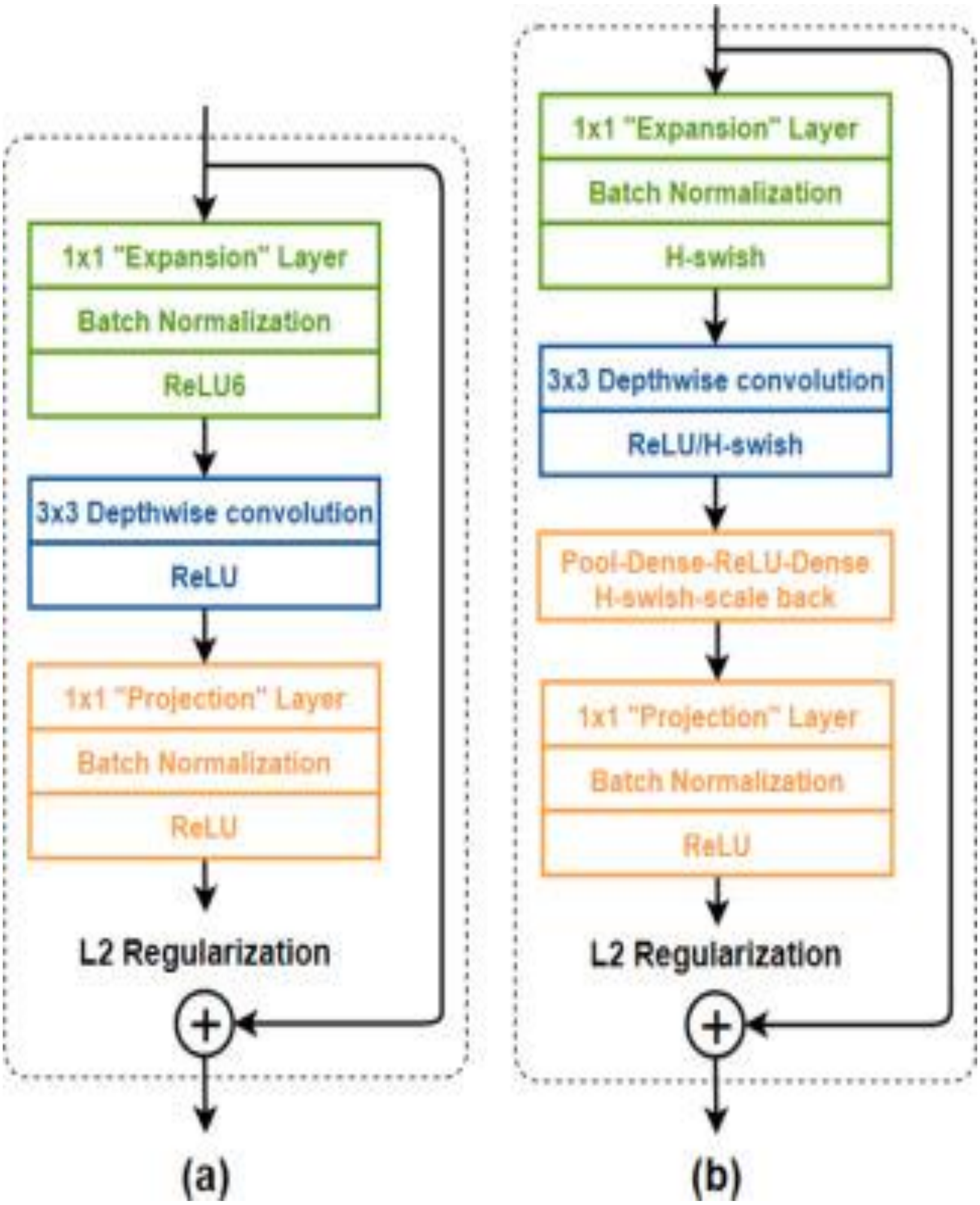


Fig 2.3 Quantized MobileNet Process

2.4 ADVANTAGES OF PROPOSED SYSTEM

The proposed garbage classification system using TensorFlow Lite and MobileNet has several advantages. Firstly, training the machine learning model on Teachable Machine website is simple and requires minimal technical expertise. Secondly, the system eliminates the need for external hardware like a webcam and external image processing system like Raspberry Pi, which reduces the overall cost of the system. Additionally, the system uses an Android phone and ESP8266 microcontroller that are readily available and easy to install. The Android application can be installed on any smartphone, making it accessible to a wide range of users. The proposed system is also efficient in terms of its processing power and memory requirements, thanks to the use of MobileNet architecture. These advantages make the proposed system a cost-effective and accessible solution for garbage classification in developing countries.

2.5 DEEP LEARNING

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behaviour of the human brain – albeit far from matching its ability – allowing it to "learn" from a large amount of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit fraud detection) as well as emerging technologies (such as self-driving cars).

2.5.1 Convolutional Neural Network

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets can learn these filters/characteristics. Its Architecture is shown in Fig 2.4.

The architecture of a ConvNet is analogous to that of the connectivity pattern of neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

Types of CNN Models:

• LeNet

• AlexNet

• ResNet

• GoogleNet

• MobileNetV1

• ZfNet

Here are the most significant advantages of convolutional neural networks (CNNs):

- CNNs do not require human supervision for the task of identifying important features.
- They are very accurate at image recognition and classification.
- Weight sharing is another major advantage of CNNs.
- Convolutional neural networks also minimize computation in comparison with a regular neural network.
- CNNs are much more independent to geometrical transformations like Scaling, Rotation etc.

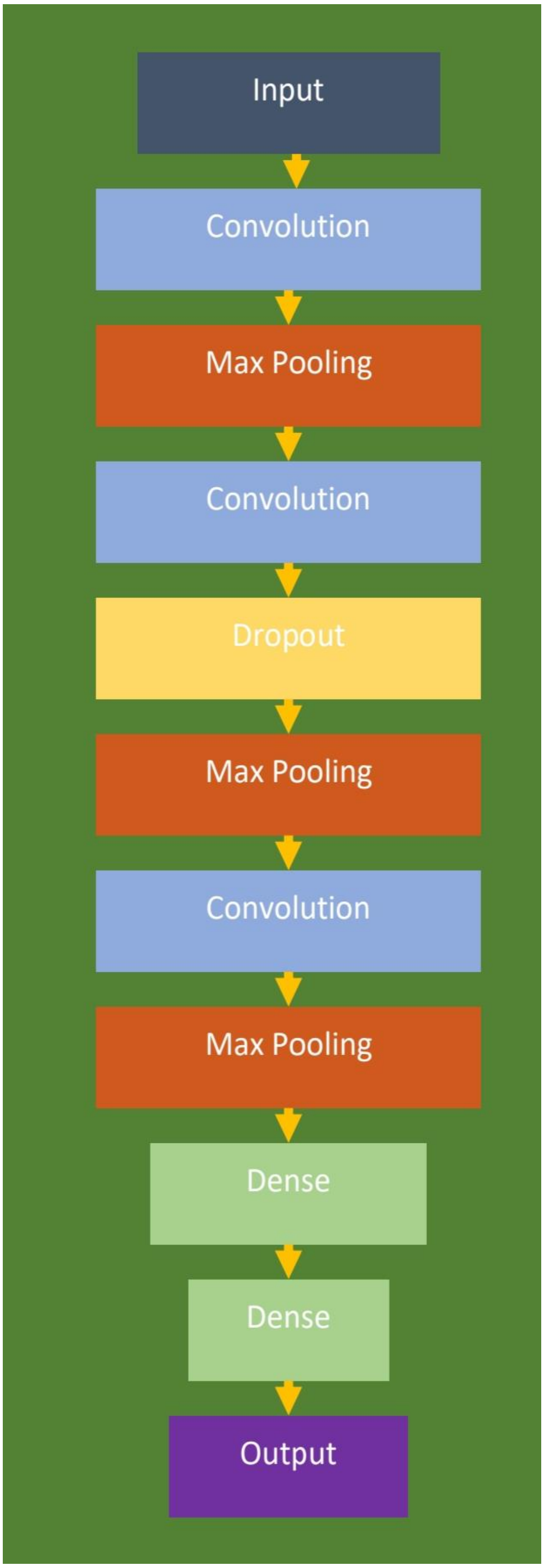


Fig 2.4 Convolutional Neural Network Architecture

2.5.2 Computer Vision

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos, and other visual inputs – and take actions or make recommendations based on that information. Different types of computer vision techniques are shown in Fig 2.5.

Computer vision needs lots of data. It runs analyses of data over and over until it discerns distinctions and ultimately recognizes images. For example, to train a computer to recognize automobile tires, it needs to be fed vast quantities of tire images and tire-related items to learn the differences and recognize a tire, especially one with no defects

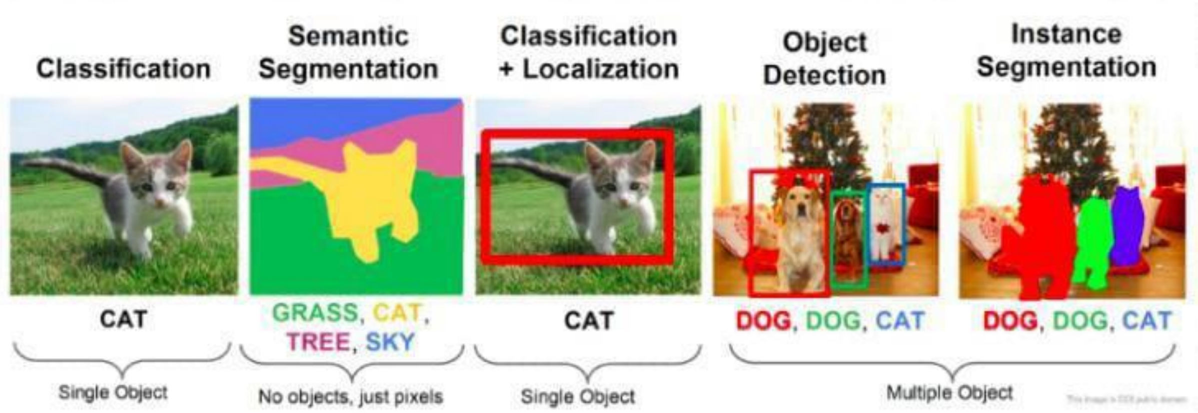


Fig 2.5 **Computer Vision techniques**

2.5.2.1 Image classification

Image classification sees an image and can classify it (a dog, an apple, a person's face). More precisely, it can accurately predict that a given image belongs to a certain class. For example, a social media company might want to use it to automatically identify and segregate objectionable images uploaded by users.

2.5.2.2 Object Detection

Object Detection can use image classification to identify a certain class of images and then detect and tabulate its appearance in an image or video. Examples include detecting damage to the assembly line or identifying machinery that requires maintenance. In our project, we used this Object Detection method to sort the plastics from other materials.

2.5.2.3 Object Tracking

Object tracking follows or tracks an object once it is detected. This task is often executed with images captured in sequence or real-time video feeds. Autonomous vehicles, for example, need to not only classify and detect objects such as pedestrians, other cars, and road infrastructure, they need to track them in motion to avoid collisions and obey traffic laws. This is depicted in Fig 2.6.

Fig 2.6 **Object Tracking**

2.5.2.4 Semantic Segmentation

Semantic Segmentation is a Deep learning algorithm that associates a label or category with every pixel in an image. It is used to recognize a collection of pixels that form distinct categories. For example, an autonomous vehicle needs to identify vehicles, pedestrians, traffic signs, pavement, and other road features. Semantic Segmentation is used in many applications such as automated driving, medical imaging, and industrial inspection.

U-Net is a semantic segmentation technique originally proposed for medical imaging segmentation.

2.5.2.5 Instance Segmentation

Instance Segmentation is identifying each object instance for every known object within an image. Instance segmentation assigns a label to each pixel of the image. It is used for tasks such as counting the number of objects. Instance Segmentation is identifying each object instance for every known object within an image. Instance segmentation assigns a label to each pixel of the image. It is used for tasks such as counting the number of objects.

2.6 TensorFlow Lite

TensorFlow Lite is a lightweight version of the popular machine learning framework TensorFlow. It is designed to be used on mobile and embedded devices with limited computational resources, such as smartphones, IoT devices, and microcontrollers. Its architecture is shown in Fig 2.7 and Fig 2.8 shows how it is deployed. TensorFlow Lite allows developers to run machine learning models on these devices with lower latency and smaller memory footprint. It supports various hardware accelerators like GPU, DSP, and NPU, which enable faster inference and better performance. TensorFlow Lite also provides a set of tools for model conversion, optimization, and deployment.

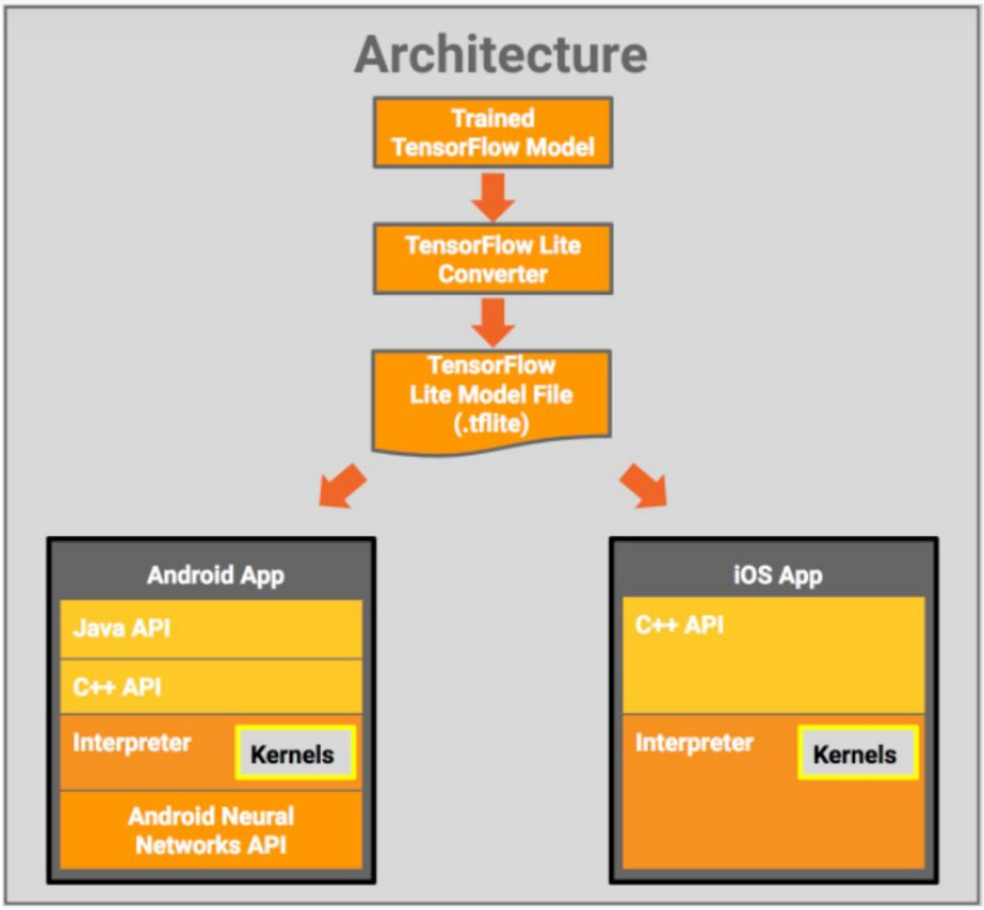


Fig 2.7 Architectural Design of TensorFlow Lite model

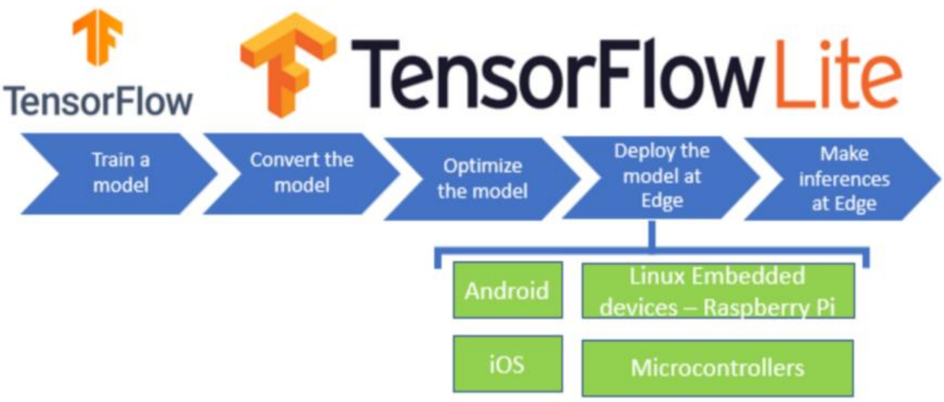


Fig 2.8 TensorFlow Lite Process

CHAPTER 3

METHODOLOGY

3.1 HARDWARE

3.1.1 ESP8266

ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability designed for the Internet of Things (IoT) applications. It was first introduced by the Chinese company Espressif Systems in 2014 and quickly gained popularity among the hobbyist and maker community due to its low cost, ease of use, and extensive community support. It is shown in Fig 3.1.

The ESP8266 has a built-in Wi-Fi module, which allows it to connect to a Wi-Fi network and communicate with other devices over the internet. It also has a built-in microcontroller, which can be programmed to perform various tasks and interact with sensors, actuators, and other components.

The ESP8266 can be programmed using the Arduino IDE or other development environments, and there are many libraries and examples available online to help developers get started quickly. Its low cost and ease of use make it an ideal platform for prototyping and experimenting with IoT applications, such as home automation, smart appliances, and remote monitoring.

NodeMCU ESP8266 has several pins, each with a specific function. The board has 11 digital input/output pins and one analog input pin. These pins can be used to connect and control various sensors, actuators, and other electronic components. Additionally, there are several power pins, including the 5V, 3.3V, and GND pins, which provide the necessary voltage and ground for the board and the connected components. The pin diagram is shown in Fig 3.2.

Fig 3.1 NodeMCU ESP8266

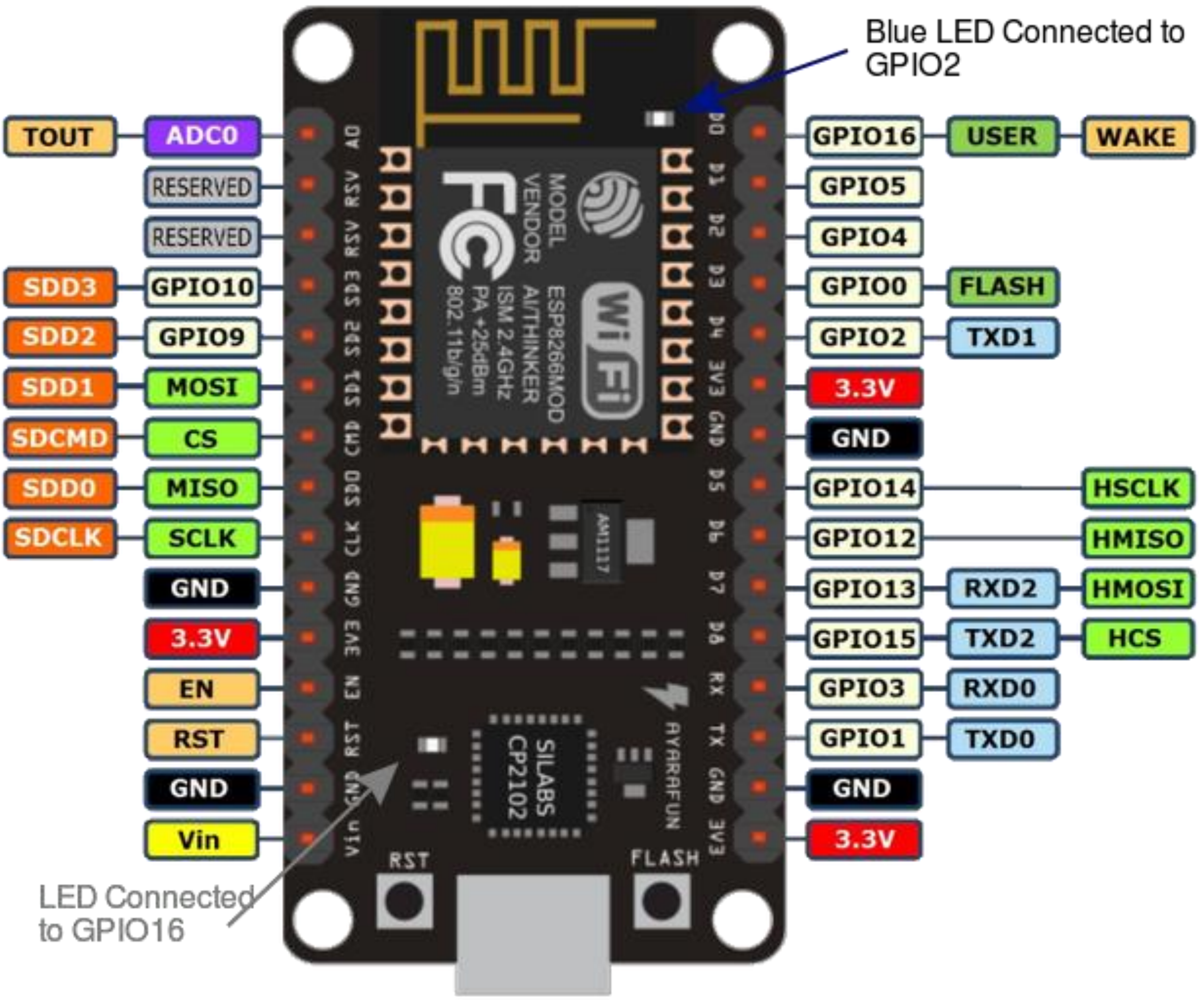


Fig 3.2 NodeMCU EPS8266 Pin diagram

3.1.2 Servo Motor

We use MG995 servo motor in this project. A servo motor is self-contained electrical device which is used for rotation of part of the machine with great precision. The output shaft of this motor can be moved at a particular angle or position or direction according to our requirements. Our prototype uses Servo motor for sorting the waste into their respective bins. The Servo is attached with a fibre board and this board is made as a tilt plate. This tilt plate is responsible for moving the waste into their respective bins. The servo motor and its pin diagram are shown in Fig 3.4 and Fig 3.3 respectively.

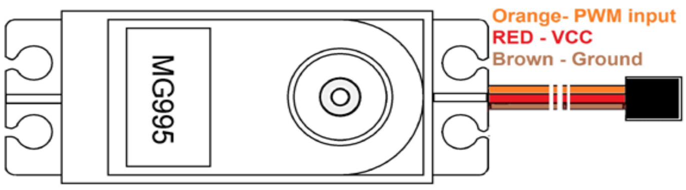
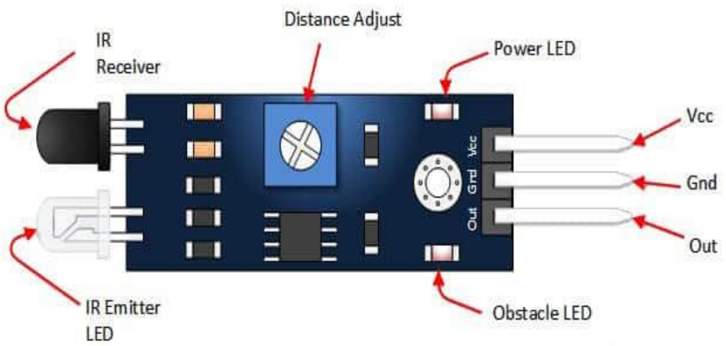


Fig 3.3 Servo Motor pin Diagram

Fig 3.4 Servo Motor

3.1.3 IR Sensor

We use an active infrared sensor in this project. It contains an IR LED and an IR photodiode. When an object is within the range of the sensor, the IR light from the LED gets reflected on the surface of the object and detected by the photodiode which triggers the sensor. It includes a potentiometer to adjust the sensitivity of the sensor. Our prototype uses this sensor to detect the garbage objects traveling on the conveyor. When it detects one, it triggers the microcontroller to stop the conveyor. This sensor is placed below the phone, so that the object is visible to the camera. The IR Sensor and its pin diagram are shown in Fig 3.6 and Fig 3.5 respectively.



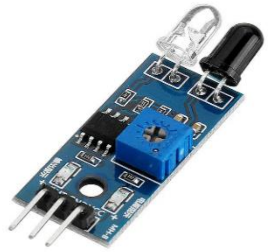


Fig 3.5 IR Sensor Pin Diagram

Fig 3.6 IR Sensor

3.1.4 Plywood

Plywood is used as a building material to construct the conveyor system and tilt plate due to its strength, durability, and ease of customization.

3.1.5 DC Gear Motor

A DC Geared Motor is a type of motor that uses direct current (DC) to drive its motion. It is designed with a gear system that allows for the motor to produce a high torque output, which makes it ideal for use in applications that require a significant amount of force. In the proposed system, the DC Geared Motor is used to drive the conveyor that moves the garbage along the sorting process. By using a motor with a high torque output, the conveyor can move heavy loads of garbage efficiently and quickly. Additionally, the speed of the conveyor can be easily controlled using a microcontroller, which allows for precise and accurate movement of the garbage through the sorting process. A DC gear motor is shown in Fig 3.7.

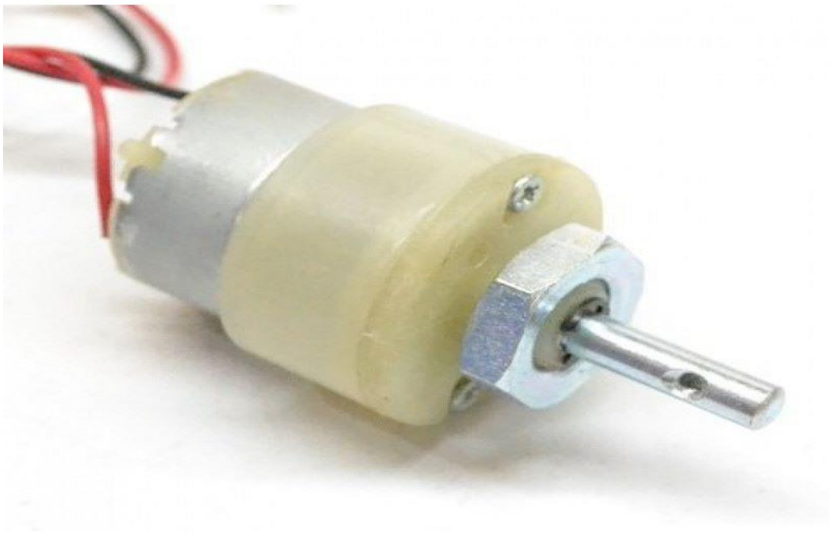


Fig 3.7 DC Gear Motor

3.1.6 Buck Converter

A buck converter, also known as a step-down converter, is an electronic device that converts a higher DC voltage to a lower DC voltage. It is shown in Fig 3.8. In this project, two buck regulators are used to regulate the voltage levels for different components.

The first buck regulator provides a 9V output voltage, which is used to power the conveyor motor. The second buck regulator provides a 5V output voltage, which is used to power the servo motor and the ESP8266 microcontroller. The buck regulators are essential in ensuring that the components receive a stable and regulated voltage supply, which is necessary for their proper functioning.

The buck converter works by using a switch to rapidly connect and disconnect the input voltage to an inductor. When the switch is closed, the inductor stores energy from the input voltage. When the switch is opened, the inductor releases the stored energy to the output circuit. A filter capacitor is used to smooth out the output voltage, resulting in a stable DC voltage at the output.

Overall, the use of buck regulators in this project ensures that the components receive the appropriate voltage levels required for their operation. This improves the overall efficiency and reliability of the system.

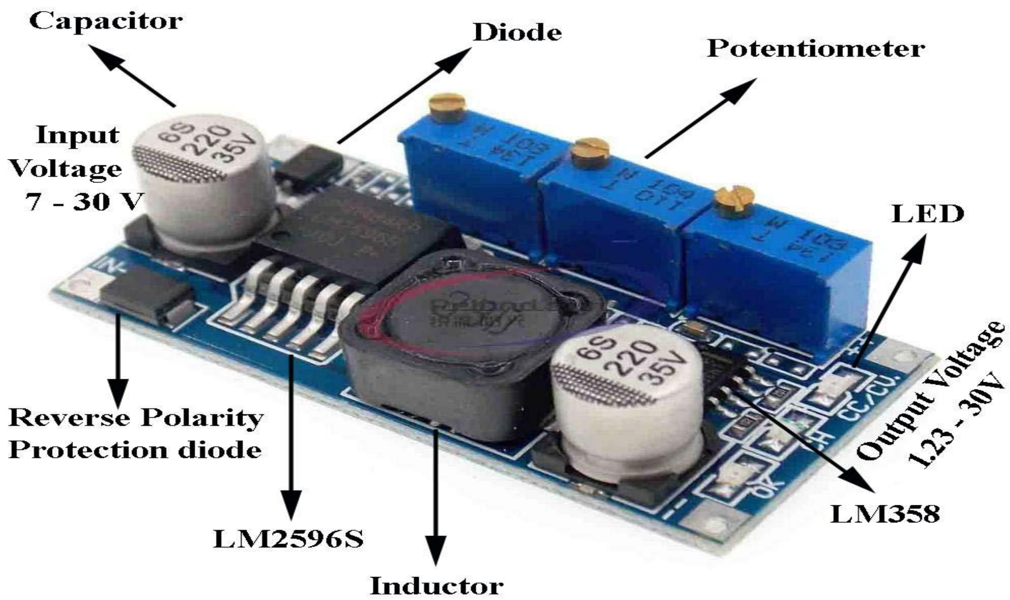


Fig 3.8 Buck Converter Diagram

3.1.7 Motor Driver

The L298N is a dual H-bridge motor driver integrated circuit that is commonly used to drive DC motors and stepper motors. It has the ability to control the direction and speed of two motors independently and can handle a maximum current of 2A per channel. In this project, the L298N motor driver is used to drive the conveyor motor. The Motor Driver (L298N) is demonstrated in Fig 3.9.

The L298N module consists of two H-bridge circuits that can drive two DC motors independently, or one stepper motor with the ability to control both the speed and direction. The motor driver module takes a DC voltage between 7V to 35V as input and provides two DC motor outputs. The module also has a built-in voltage regulator that can provide a regulated 5V output, which is used to power the ESP8266 microcontroller.

The L298N module is connected to the NodeMCU microcontroller board, which provides the necessary signals to control the motor speed and direction. The NodeMCU sends the appropriate signals to the L298N module to control the conveyor motor. The motor driver is connected to a 12V power supply, which is used to power the conveyor motor.

The L298N motor driver is useful in this project because it provides a simple and efficient way to control the conveyor motor. It allows the user to control the speed and direction of the motor, which is important in a waste segregation system where the conveyor speed needs to be adjusted based on the type and amount of waste being processed. The L298N module is also very affordable and readily available, making it a popular choice for hobbyists and DIY projects.

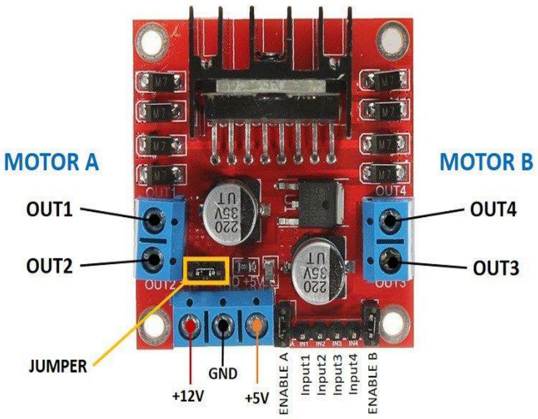


Fig 3.9 Motor Driver (L298N) Diagram

3.2 SOFTWARE

3.2.1 Android Studio

Android Studio is a popular integrated development environment (IDE) used for building Android apps. Android Studio is presented in Fig 3.10. It provides developers with a range of tools and features that make it easy to create, debug, and modify applications. The platform offers an intuitive interface that simplifies the development process and is well-suited for both beginners and experienced developers.

One of the main advantages of Android Studio is its ability to simplify the coding process. It includes features like code completion and syntax highlighting that make it easy to write code quickly and without errors. It also includes a powerful debugger that allows developers to quickly identify and fix issues in their code.

In addition, Android Studio includes a number of tools for testing and refining applications. This includes support for emulators and real devices, as well as tools for analysing and profiling code. The platform also supports version control systems like Git, making it easy to collaborate with other developers and track changes to code over time.

Finally, Android Studio is used for creating APK files, which are the package files used to distribute and install Android applications. It includes tools for signing and packaging applications, as well as support for publishing applications to the Google Play Store.

Overall, Android Studio is a powerful and versatile tool that is essential for anyone looking to develop Android applications. Its intuitive interface, powerful debugging tools, and support for testing and publishing make it a valuable tool for developers at all levels of experience.

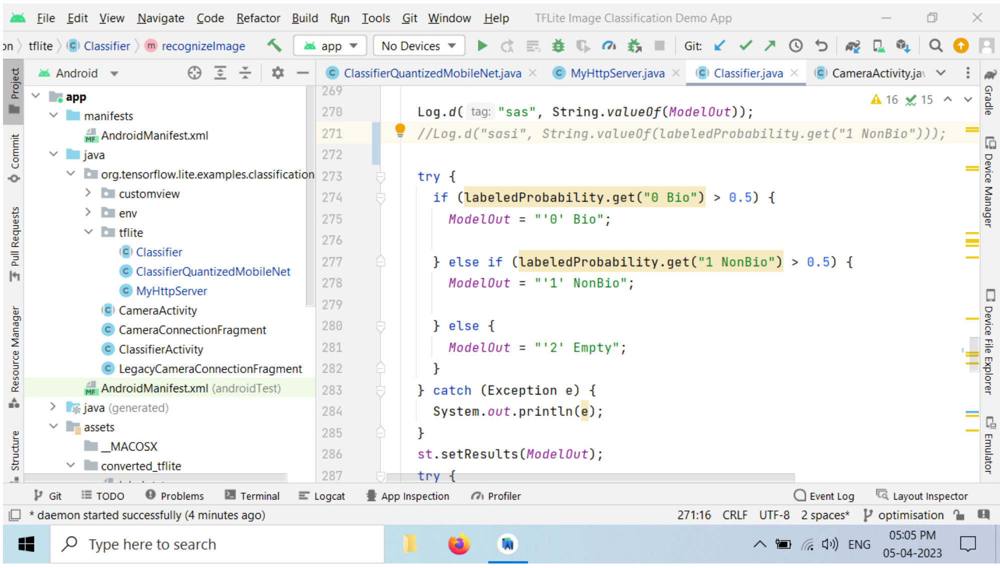


Fig 3.10 Android Studio

3.2.2 Arduino IDE

The Arduino Integrated Development Environment (IDE) is an open-source platform used for programming and developing electronics projects. The IDE provides a user-friendly interface for writing and uploading code to Arduino boards, including the ESP8266 microcontroller used in this project. The Arduino IDE is shown in Fig 3.11.

To program the ESP8266, the Arduino IDE must be configured to recognize the ESP8266 board. The board can then be selected in the IDE and programmed using the Arduino programming language. The ESP8266 can also be programmed using the Lua programming language, but this was not used in this project.

The Arduino IDE includes a serial monitor that can be used for debugging and testing the ESP8266. The serial monitor displays the data sent and received by the microcontroller, allowing developers to monitor and debug their code. This is especially useful when testing the communication between the ESP8266 and the Android phone, as it allows developers to view the data being sent and received in real-time.

Additionally, the Arduino IDE includes a library manager that allows developers to easily add and manage libraries for their projects. This is useful for adding libraries that provide additional functionality to the ESP8266, such as libraries for controlling servo motors or communicating with sensors.

In summary, the Arduino IDE is a powerful and versatile tool for programming microcontrollers like the ESP8266. It provides a user-friendly interface for writing and uploading code, includes a serial monitor for debugging and testing, and has a library manager for adding additional functionality to projects.

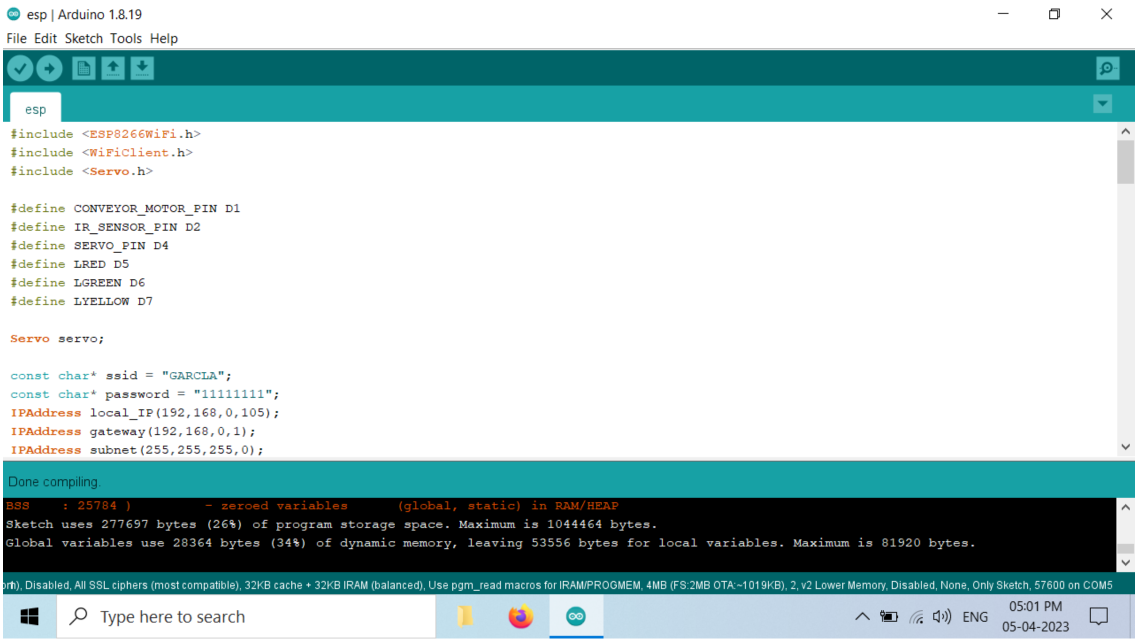


Fig 3.11 Arduino IDE

3.2.3 GitHub

GitHub is a web-based platform that serves as a repository for source code and other software development projects. It has become an essential tool for developers to manage their projects and work collaboratively with others. One of the key features of GitHub is the ability to track changes to a project over time using a version control system. This allows developers to discard unwanted changes and revert to earlier versions of the code, making it easier to manage and maintain code. GitHub is illustrated in Fig 3.12.

GitHub is also useful for sharing code and collaborating with team members during the development process. Therefore, we created a private repository on GitHub. Developers can create a repository and invite others to contribute, making it easy to share code and work on projects together. The platform includes features such as issue tracking, pull requests, and code reviews, which make it easier to manage projects and ensure that code changes are reviewed and approved before being merged into the main branch.

Additionally, GitHub provides a platform for developers to showcase their work, making it easy for potential employers or collaborators to view their code and projects. It also allows developers to contribute to open-source projects and collaborate with others on a global scale. Android application of our project builds upon a foundational codebase sourced from tensorflow/examples GitHub repository, which has since been substantially tailored to our unique specifications.

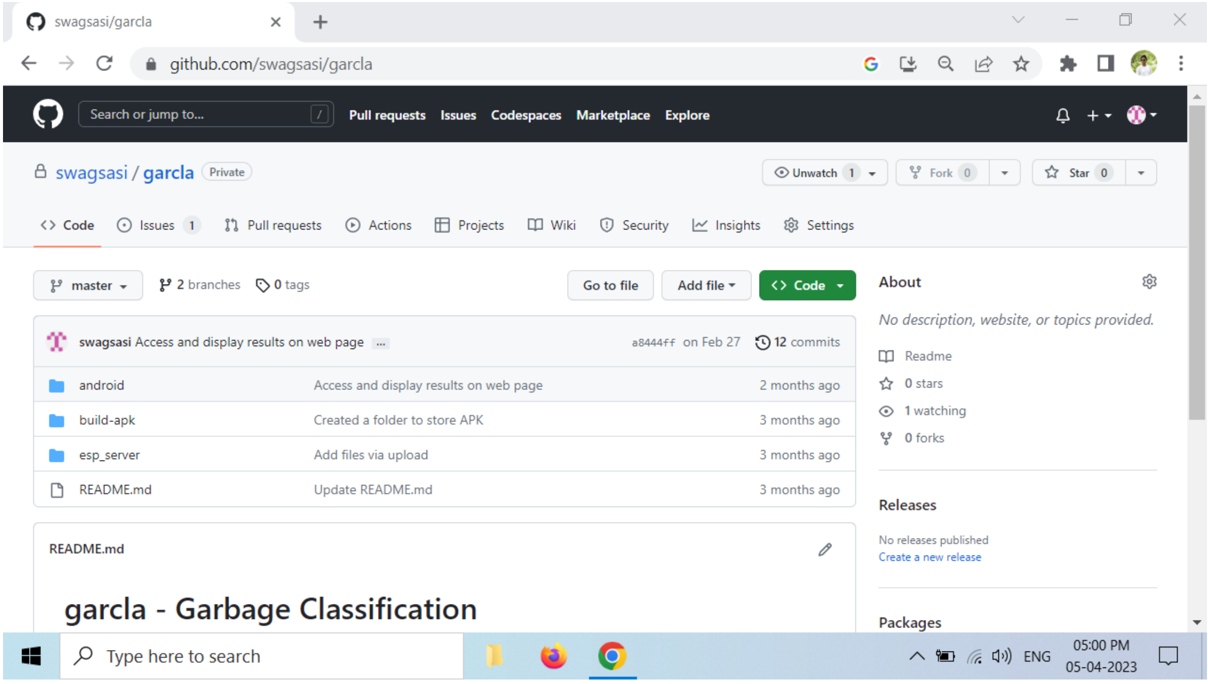


Fig 3.12 GitHub

3.2.4 Teachable Machine

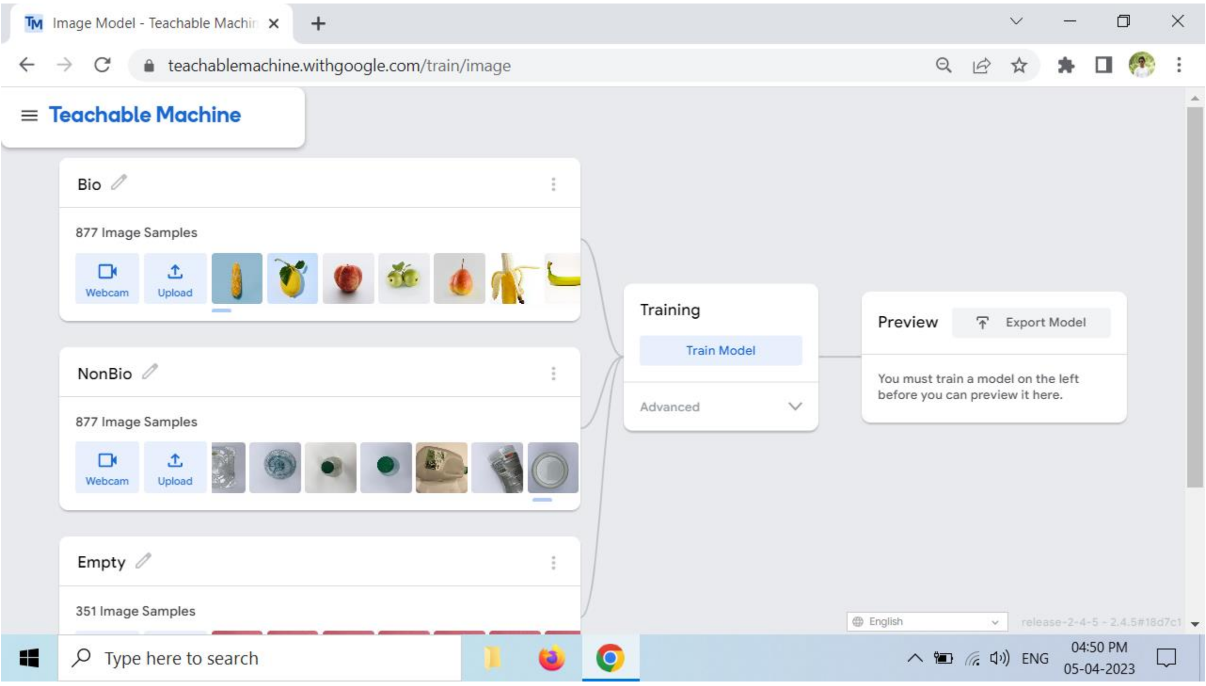
Teachable Machine is a web-based tool developed by Google that allows users to create custom machine learning models without the need for extensive coding knowledge. It is a simple and user-friendly platform that enables individuals to train their own image, audio, or pose recognition models by uploading and categorizing examples of data. Fig 3.13 presents Teachable Machine website.

Teachable Machine uses a neural network to process data and generate a model that can be exported in different formats, such as TensorFlow or TensorFlow Lite. This makes it easy to integrate the trained model into various applications, including web or mobile-based platforms.

One of the key advantages of Teachable Machine is its accessibility. It allows anyone with an internet connection to train their own machine learning model, regardless of their technical background. Additionally, Teachable

Machine is open-source and free to use, making it an ideal platform for small-scale projects or educational purposes.

Overall, Teachable Machine is a valuable tool for individuals or organizations seeking to implement machine learning into their projects or applications, but lack the resources or expertise to build their own models from scratch.



3.13 Teachable Machine

3.3 DATA COLLECTION AND PREPROCESSING

In the field of deep learning, the first and foremost step is dataset preparation and it is also the most hectic process. It is a long time-consuming process, where most people do many mistakes and become frustrated with their work. The quality of the dataset will decide the performance of the model. The higher the quality the higher will be the performance. To make the dataset quality there are many data augmentation techniques available. There is a step-by-step procedure for creating a clean and quality dataset. They are,

- Data Collection
- Data Cleaning
- Data Splitting
- Pre-processing
- Data Augmentation

3.3.1 Data Collection

Collecting data for training the ML model is the basic step in the machine learning pipeline. The predictions made by ML systems can only be as good as the data on which they have been trained. Data collection is demonstrated in Fig 3.14.

Following are some of the problems that can arise in data collection:

- Inaccurate data. The collected data could be unrelated to the problem statement.
- Missing data. Sub-data could be missing. That could take the form of empty values in columns or missing images for some class of prediction.
- Data imbalance. Some classes or categories in the data may have a disproportionately high or low number of corresponding samples. As a result, they risk being under-represented in the model.
- Data bias. Depending on how the data, subjects, and labels themselves are chosen, the model could propagate inherent biases on gender, politics, age, or region, for example. Data bias is difficult to detect and remove.

Several techniques can be applied to address those problems:

- Pre-cleaned, freely available datasets. If the problem statement (for example, image classification, object recognition) aligns with a clean, preexisting, properly formulated dataset, then take advantage of existing, open-source expertise.

• Web crawling and scraping. Automated tools, bots, and headless browsers can crawl and scrape websites for data.

• Private data. ML engineers can create their data. This is helpful when the amount of data required to train the model is small and the problem statement is too specific to generalize over an open-source dataset.

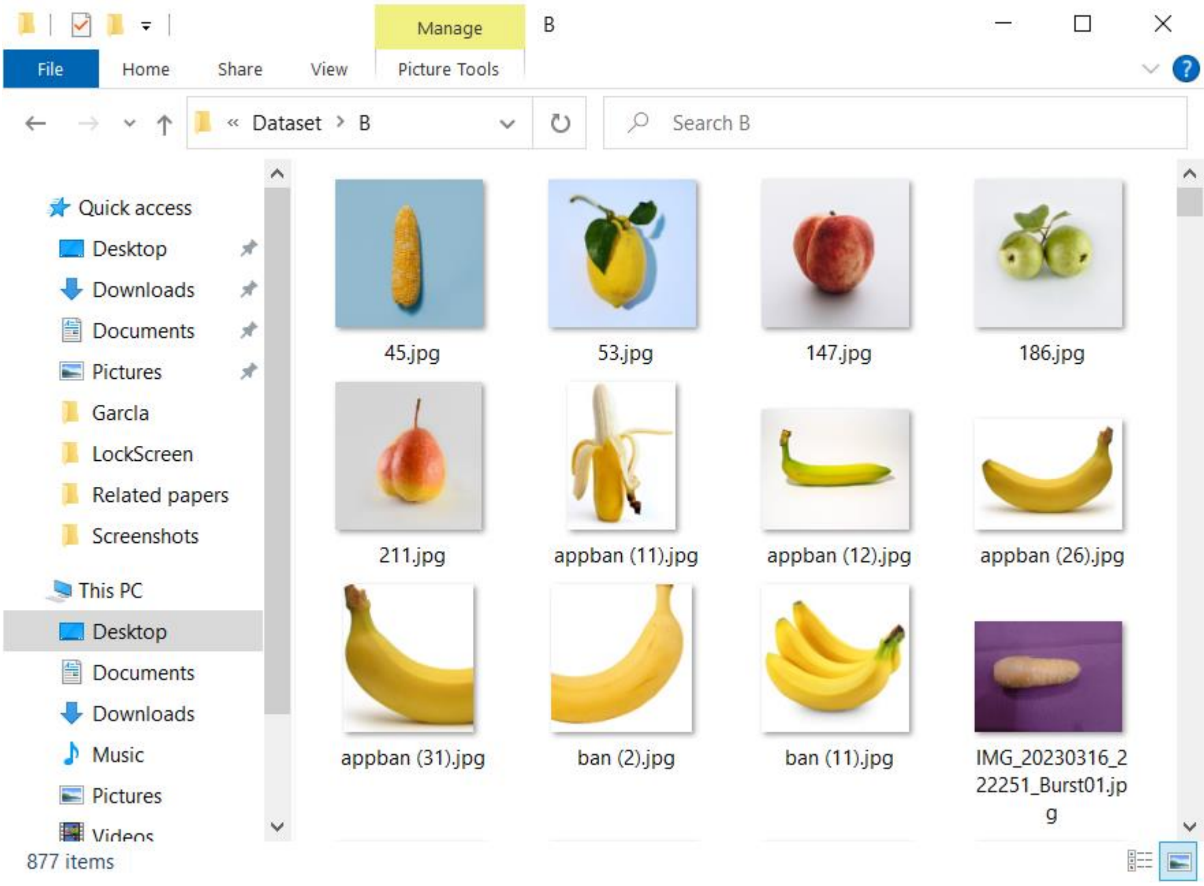


Fig 3.14 Data Collection

3.3.2 Data Cleaning

Data cleaning is the process of preparing data for analysis by removing or modifying data that is incorrect, incomplete, irrelevant, duplicated, or improperly formatted. Data cleaning is a lot of muscle work. There’s a reason data cleaning is the most important step if you want to create a data culture, let alone make airtight predictions. It involves:

- Fixing spelling and syntax errors
- Standardizing data sets
- Correcting mistakes such as empty fields
- Identifying duplicate data points

3.3.3 Data Splitting

Data splitting is when data is divided into two or more subsets. Typically, with a two-part split, one part is used to evaluate or test the data and the other to train the model. Data splitting is an important aspect of data science, particularly for creating models based on data. Here, the Roboflow website will automatically divide our datasets into Train, Valid, and Test datasets by the ratio of 70:20:1.

3.3.4 Pre-processing

Pre-processing data is a common first step in the deep learning workflow to prepare raw data in a format that the network can accept. For example, you can resize image input to match the size of an image input layer. You can also pre-process data to enhance desired features or reduce artifacts that can bias the network.

3.3.5 Data Augmentation

A convolutional neural network that can robustly classify objects even if it's placed in different orientations is said to have the property called invariance. More specifically, a CNN can be invariant to translation, viewpoint, size, or illumination (Or a combination of the above). This essentially is the premise of data augmentation. In the real-world scenario, we may have a dataset of images taken under a limited set of conditions. But our target application may exist in a variety of conditions, such as different orientations, locations, scale, brightness, etc. We account for these situations by training our neural network with additional synthetically modified data.

3.4 Training of TensorFlow Lite Model

1. **Collect and prepare the dataset**: Collect a dataset of images for both biodegradable and non-biodegradable waste items. The images should be clear and high-quality. It is recommended to have at least 100 images for each class.

2. **Upload and label the dataset**: Go to the Teachable Machine website and select the Image Project. Upload the dataset and label each image as either biodegradable or non-biodegradable.

3. **Train the model**: Click on the Train Model button to start the training process. The model will be trained using the dataset you provided. The training process may take a few minutes or more, depending on the size of the dataset.

4. **Test the model**: After training, you can test the model using the Test Model button. Upload a new image and see if the model correctly classifies it as biodegradable or non-biodegradable.

5. **Export the model**: If the model performs well, you can export it by clicking on the Export Model button. Choose the TensorFlow Lite option to export the model in a format that can be used on an Android phone.

6. **Integrate the model**: Integrate the model into your Android application using the TensorFlow Lite library. Use the model to classify waste items as biodegradable or non-biodegradable in real-time.

3.5 SYSTEM COMPONENTS

The system consists of three main components, they are tilt plate, conveyor and a control box.

3.5.1 Control box

The control box of the waste segregation system is an essential component that houses the power supply and the main controller. It contains two buck regulators, one that provides a 9V supply for the conveyor motor and the other that provides a 5V supply for the servo. The NodeMCU acts as the main controller for the system, and its functions are indicated by three LED lights - yellow, green, and red - which show the current state of the system. The motor driver used in the system is L298N, which helps to control and drive the conveyor motor. The LED present in the control box provides visual cues to the user about the system's performance and progress. The control box is an important part of the waste segregation system, as it provides the necessary power and control to run the system and helps to segregate waste more efficiently. The contents of the control box are shown in Fig 3.15.

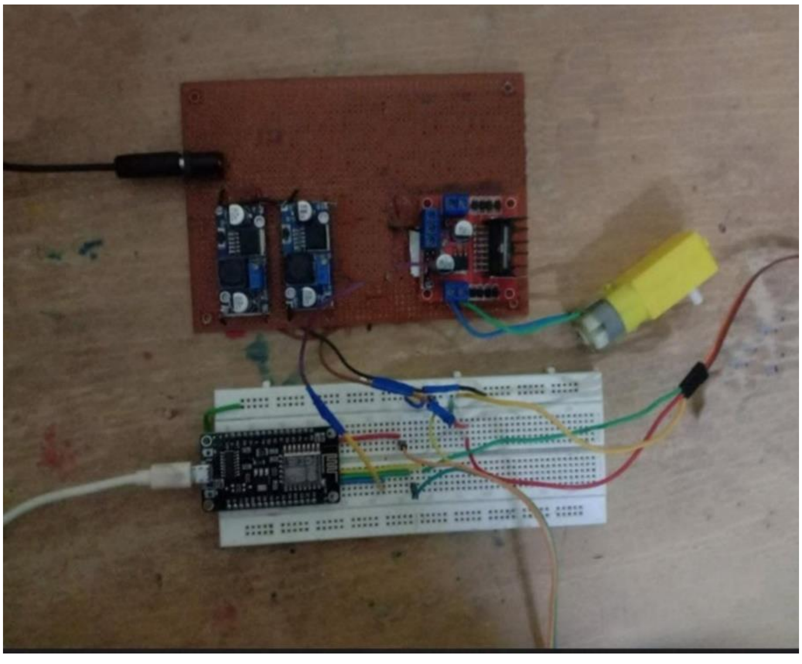


Fig 3.15 Contents of control box

3.5.2 Conveyor System

The conveyor system is a crucial component of the waste segregation system. It is made out of plywood and PVC pipe, which are lightweight and durable materials. The conveyor is designed in an 'H' shape and is supported by bearings on each side of the roller. The roller is connected to the motor shaft, which is powered by the control box. The conveyor belt is made out of TT belt material, which has a length of 900 mm and a width of 170mm. An IR sensor is mounted on a platform to detect the waste materials passing through the conveyor system. The conveyor is placed on top of the control box to save space and facilitate the efficient flow of waste materials. Overall, the conveyor system is an essential component of the waste segregation system as it helps transport the waste materials to the tilt plate for further segregation. Fig 3.16 illustrates the Conveyor System.

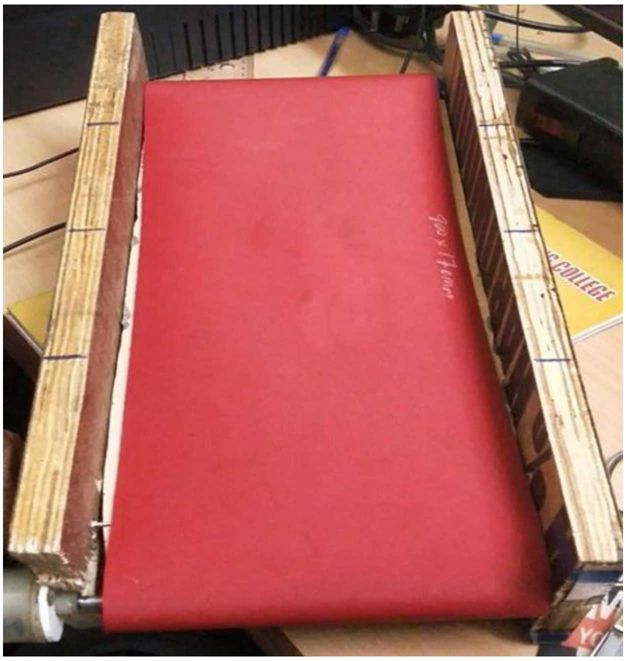


Fig 3.16 Conveyor System

3.5.3 Tilt Plate System

The tilt plate system is an important component of the proposed waste segregation system, as it allows for the separation of biodegradable and non-biodegradable waste. The plate is made out of fibre board, which is a strong and lightweight material that can withstand the weight of the waste materials. It is supported on both sides by plywood, with one side acting as a bearing and the other side connected to the servo motor. The servo motor is responsible for controlling the movement of the tilt plate. When the waste material passes under the IR sensor, it sends a signal to the microcontroller which then activates the servo motor to tilt the plate to the appropriate angle. This causes the waste material to slide off the plate and into the designated bin. The use of the tilt plate system ensures that the waste materials are properly separated, making the waste segregation process more efficient and accurate. Additionally, the use of a servo motor allows for precise control over the angle of the tilt plate, ensuring that the waste materials are correctly sorted. Tilt Plate is presented in Fig 3.17.

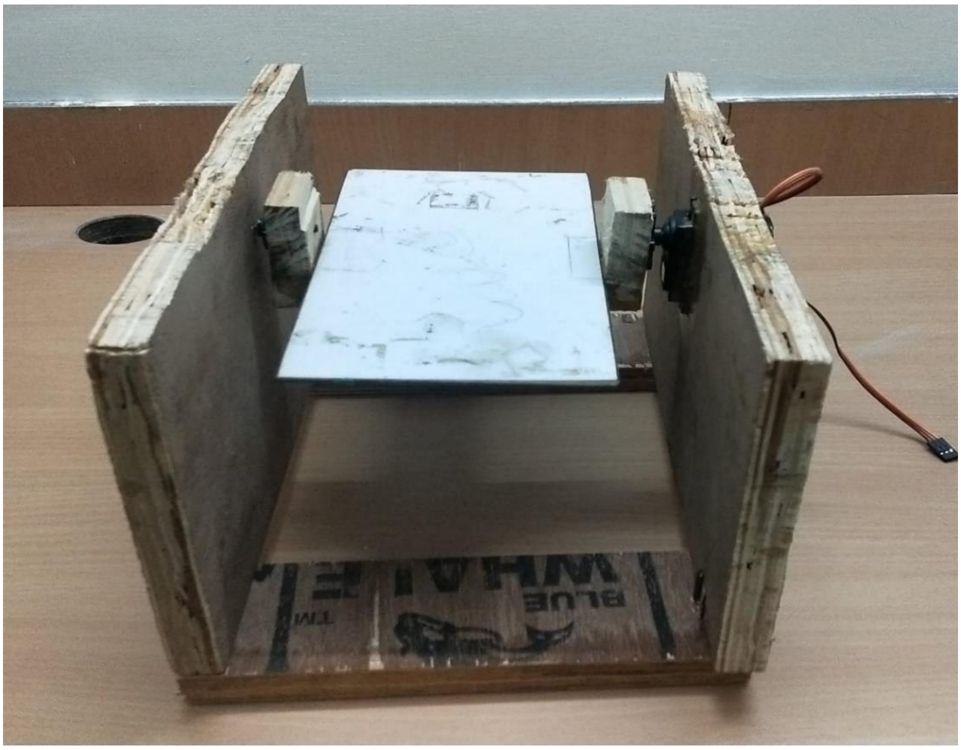


Fig 3.17 Tilt Plate

3.3 INTEGRATION OF HARDWARE AND SOFTWARE

The integration of machine learning with hardware has opened up new possibilities for solving real-world problems. One such problem is the challenge of garbage segregation in cities. In this paper, a low-cost solution using TensorFlow Lite machine learning model embedded in an Android application is proposed. This eliminates the need for expensive external image processing hardware

Once the power supply is provided to the system, the 12V conveyor motor will begin running. The IR sensor, also fixed to the conveyor motor, will detect any objects that pass in front of it. When an object is detected, the IR sensor will send an output signal to the ESP8266 microcontroller, which will then send a command to the motor driver to halt the power supply to the conveyor motor. Circuit diagram is shown in Fig 3.19.

At this point, the object will be stationary in front of the mobile phone fixed above the conveyor. The mobile phone is equipped with a machine learning model from Tensor Flow Lite, which has been installed as an app on the phone. This machine learning model will be used to scan the object below it.

In the proposed waste classification system, the data flow starts with the camera capturing the image of the waste object. The image is then sent to the Android app installed on the phone. The Android app preprocesses the image to resize and normalize it, and then passes it to the TensorFlow Lite model for classification. The model then outputs a prediction for the waste object, indicating whether it is biodegradable or non-biodegradable.

The prediction value is stored in a variable, which is then used by the MyHttpServer instance running on the phone. When an HTTP request is received by the phone, the MyHttpServer generates a response based on the prediction value. This response is sent back to the ESP8266 microcontroller, which is connected to the phone via Wi-Fi.

After getting the response from the App either as 0 or 1, the ESP8266 will give a command to the servo motor fixed with the tilt plate, which will turn accordingly like for bio 0° and for non-bio 180° rotation. Then the conveyor motor will start to run and the object on the conveyor will drop on the tilt plate. After 5 seconds of delay, another object will get detected by the IR sensor and the process will continue again as a cycle. Fig 3.18 demonstrates the flow chart of the full process.

In manual mode, a user can take a picture of the garbage object using the Android application and receive the classification result.

In automatic mode, the garbage object is automatically detected by the IR sensor placed beneath the camera and the conveyor stops. The image is then captured and classified by the machine learning model, and the result is sent back to the microcontroller, which tilts the plate to the left or right to sort the garbage into the appropriate bin. The conveyor then starts moving again to sort the next garbage object.

Therefore, The integration of hardware and software is crucial for the proper functioning of the waste segregation system. The hardware components are controlled by the software, which receives input from them and uses the machine learning model to generate output commands. This integration ensures accurate waste segregation based on the model's predictions.

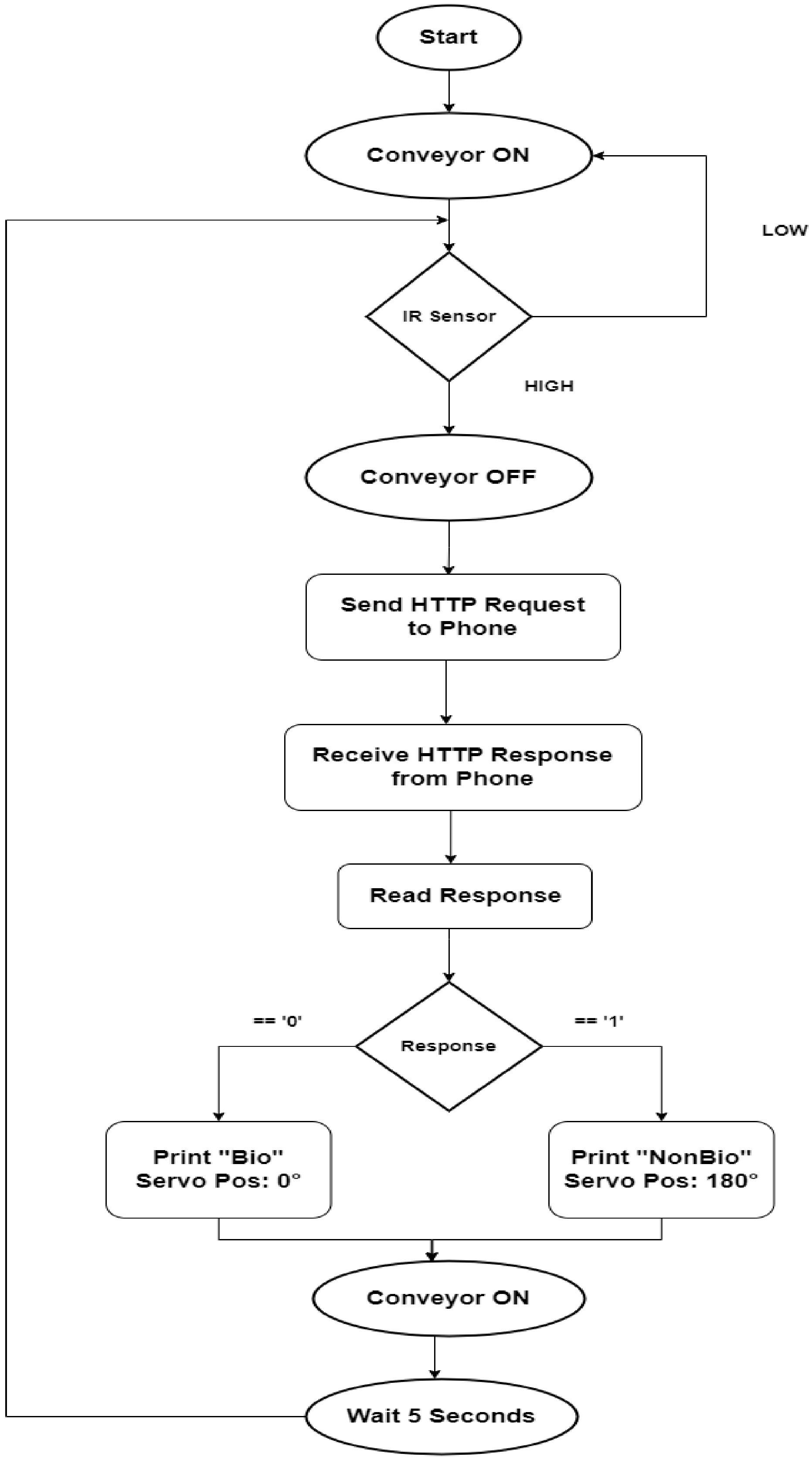


Fig 3.18 Flow Chart of Full Process

Circuit Diagram:

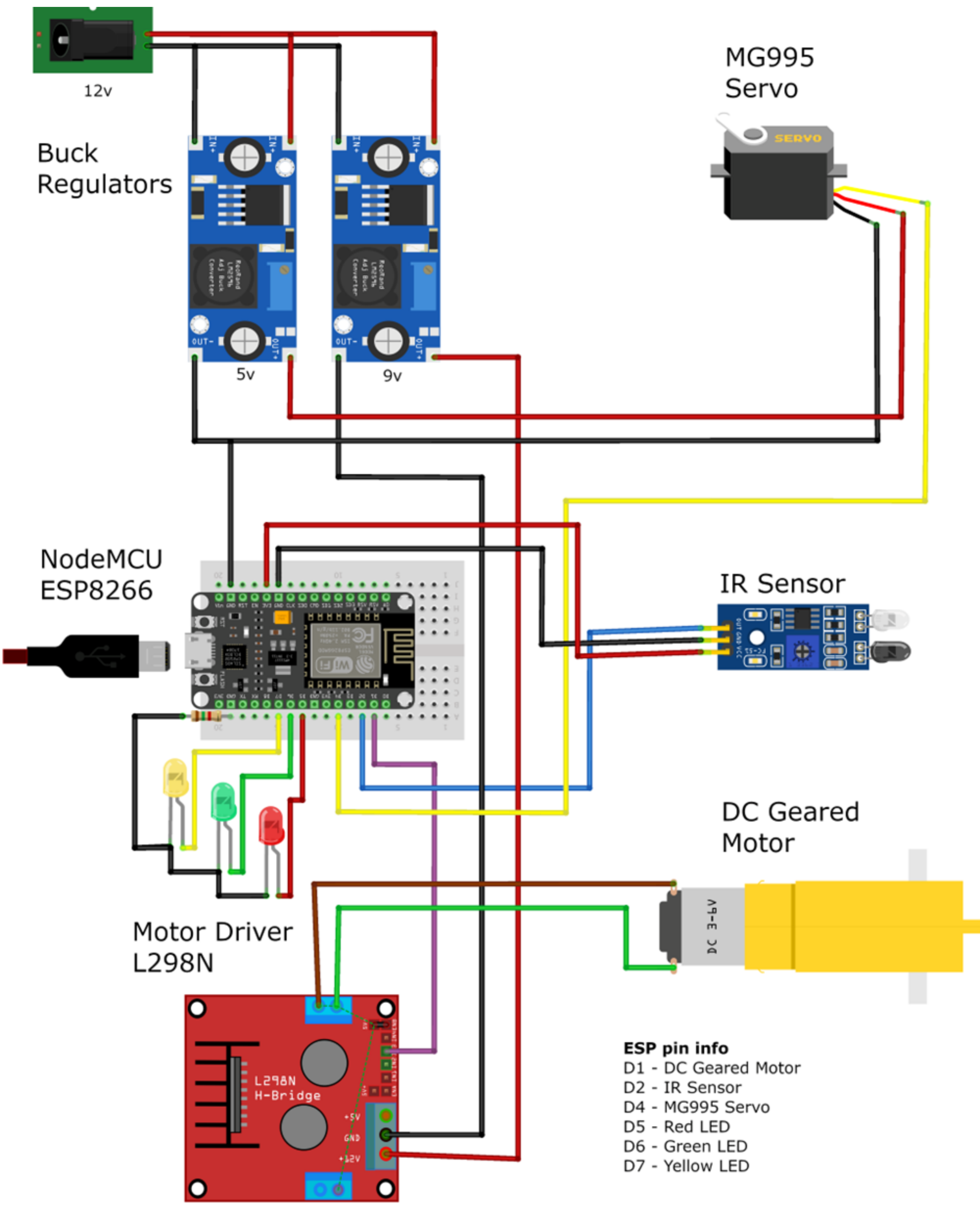


Fig 3.19 Circuit Diagram

CHAPTER 4

RESULT AND DISCUSSION

4.1 MODEL OUTPUT

After generating the APK file from Android Studio, it can be easily installed on the Android phone. Once the installation is complete, the user can turn on the phone's Wi-Fi and connect it to the ESP's network. The mobile application is then launched and placed on top of the conveyor belt, in a position where the camera can capture the objects as they pass by.

The app receives the live video stream from the phone's camera and displays it on the screen. In addition, the app also utilizes the machine learning model that has been installed using TensorFlow Lite. The model is designed to classify the objects passing on the conveyor belt into two categories, biodegradable and non-biodegradable.

The .tflite model file contains a pre-trained machine learning model, which has been optimized for mobile devices. The model is loaded into the app and is used to process the live video stream from the camera. The model's internal workings involve several convolutional and pooling layers, which are used to extract features from the images.

The extracted features are then passed through several fully connected layers, which perform the actual classification of the objects. Once the classification is complete, the app displays the results on the screen, indicating whether the object is biodegradable or non-biodegradable.

Overall, the mobile application provides a simple and intuitive way to classify objects passing on the conveyor belt, using a pre-trained machine learning model that has been optimized for mobile devices. The application can be easily installed on any Android phone and can be used to process the live video stream from the phone's camera, making it a highly convenient and versatile tool for waste management and recycling. Application output is presented in Fig 4.1.

Fig 4.1 Application Output

4.2 SYSTEM OUTPUT

To turn on the system, a micro-USB is connected to the NodeMCU board, and a 12-volt power supply is connected to the main power outlet. Once powered up, the conveyor belt starts to rotate, and the servo motor goes to the 90-degree position. The user can then place the object to be sorted onto the conveyor belt. As the object moves along the conveyor belt, it passes an IR sensor that triggers a signal to the ESP8266 microcontroller, which then turns off the conveyor motor.

At this point, the microcontroller sends an HTTP request to the Android phone that acts as a server. While the microcontroller waits for a response from the phone, it turns on the yellow LED as shown in Fig 4.3 to indicate that a request has been sent. The response from the phone is either a '0' or '1', indicating whether the object is biodegradable or non-biodegradable, respectively. Fig 4.2 shows the entire prototype.

If the response is '0', the microcontroller turns on the green LED as demonstrated in Fig 4.4, indicating that the object is biodegradable, and turns the servo motor to the 0-degree position. Conversely, if the response is '1', the microcontroller turns on the red LED, indicating that the object is non-biodegradable, and turns the servo motor to the 180-degree position. Fig 4.5 shows the system receiving 'Non-Bio' response.

After this, the conveyor belt is turned on again, and the IR sensor value is ignored for 5 seconds. This allows the object to move forward and fall down onto a tilt plate that directs it to the appropriate bin. The system is now ready to sort the next object, and the cycle repeats.

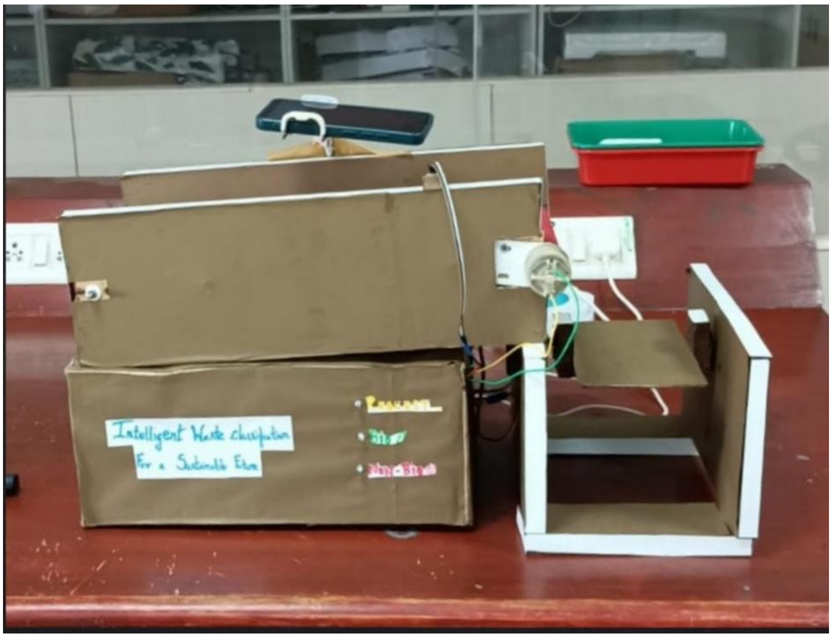


Fig 4.2 Entire Prototype

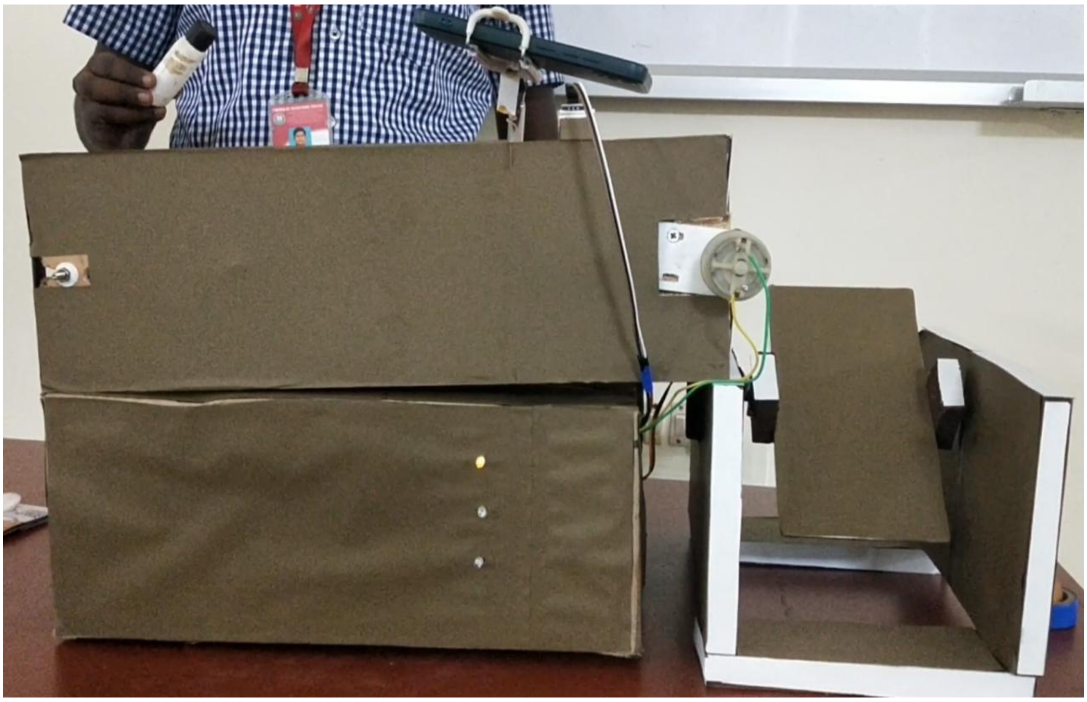


Fig 4.3 System waits for response

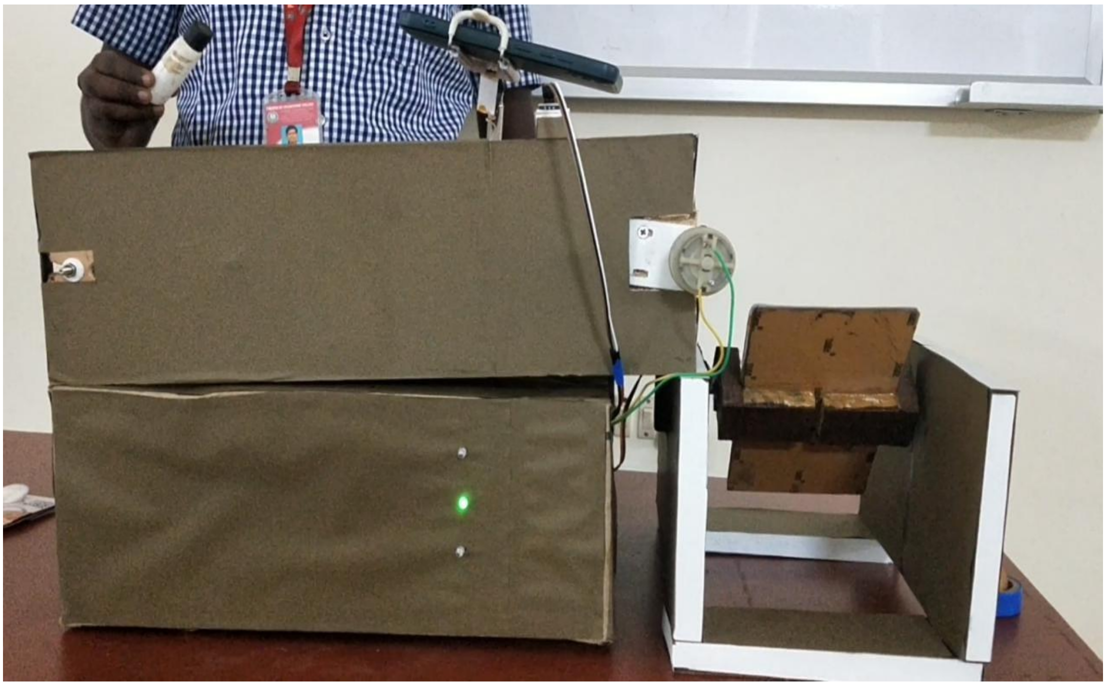


Fig 4.4 System receives ‘Bio’ response

Fig 4.5 System receives ‘Non-Bio’ response

CHAPTER 5

CONCLUSION

5.1 SUMMARY OF PROJECT'S GOAL AND OBJECTIVES

The goal of this project is to design and develop a cost-effective waste segregation system using machine learning. The system aims to address the inefficiencies and limitations of traditional garbage classification methods by automating the process and reducing the need for manual labour, which can lead to injury or disease.

To achieve this goal, the project objectives include:

- Designing and developing a hardware and software system that can effectively sort waste into biodegradable and non-biodegradable categories.
- Collecting and pre-processing a dataset of images of waste objects to train and test the machine learning model.
- Training a TensorFlow Lite machine learning model to accurately classify waste objects based on their images.
- Integrating the hardware and software components to create a functional waste segregation system.

The proposed system's scope is limited to waste segregation based on image classification using an Android mobile phone and a readily available microcontroller called ESP8266. The system eliminates the need for external image processing systems like Raspberry Pi, making it more cost-effective. Overall, the project aims to provide a practical solution to improve waste management and promote environmental sustainability.

5.2 DISCUSSION OF THE PROJECT'S CONTRIBUTIONS

The proposed waste segregation system using machine learning and low-cost hardware has several contributions to the field of waste management. Firstly, it addresses the issue of inefficient waste segregation in India, where a large portion of garbage is not properly disposed of or treated. By automating the waste segregation process, the system improves the efficiency and accuracy of waste classification, which in turn reduces the burden on manual laborers and improves their safety.

Secondly, the use of machine learning for waste segregation represents a significant step towards the application of artificial intelligence in the field of waste management. The system demonstrates the potential of machine learning in automating waste segregation and reducing the environmental impact of improper waste disposal.

Additionally, the use of low-cost hardware components makes the system accessible and affordable to communities with limited resources. The proposed system can be easily replicated and implemented in different locations, making it a scalable solution to the problem of inefficient waste segregation.

Overall, the proposed waste segregation system has the potential to contribute to the improvement of waste management practices in developing countries, reduce the negative impact of improper waste disposal on the environment, and promote the adoption of machine learning in the field of waste management.

5.3 LIMITATIONS AND RECOMMENDATIONS FOR FUTURE WORK

While the proposed waste segregation system has demonstrated promising results, there are several limitations that need to be addressed in future work.
One of the main limitations is the current system's ability to only sort waste into two categories (biodegradable and non-biodegradable). In reality, there are many different types of waste that require different forms of disposal, and the proposed system would need to be expanded to include more categories to be more effective in a real-world setting.

Another limitation is the system's reliance on an Android phone as the main hardware component. This can be a limitation as not everyone may have an Android phone, or the phone may not be powerful enough to handle the machine learning algorithms required for waste classification. Future work could explore alternative hardware options that are more accessible and can handle the computational demands of the system.

In addition, the current system relies on an IR sensor to detect when an object is on the conveyor belt. However, this approach may not be accurate enough for all types of waste. Future work could explore alternative sensors or approaches to detecting waste on the conveyor belt to improve the accuracy of the system.

Finally, the proposed system currently requires manual intervention to remove waste from the bins once they are full. A future enhancement could be to incorporate an automatic waste disposal system that can empty the bins when they are full. This could be achieved using robotic arm or a vibrating feeder
Overall, the proposed waste segregation system shows great potential for improving waste management practices, but there is still much work to be done to make it more effective and efficient in a real-world setting.

APPENDIX

1. ESP8266 Microcontroller Code

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <Servo.h>

#define CONVEYOR_MOTOR_PIN D1
#define IR_SENSOR_PIN D2
#define SERVO_PIN D4
#define LRED D5
#define LGREEN D6
#define LYELLOW D7

Servo servo;

const char* ssid = "GARCLA";
const char* password = "11111111";
IPAddress local_IP(192,168,0,105);
IPAddress gateway(192,168,0,1);
IPAddress subnet(255,255,255,0);

void setup() {
pinMode(CONVEYOR_MOTOR_PIN, OUTPUT);
pinMode(IR_SENSOR_PIN, INPUT_PULLUP);
pinMode(LRED, OUTPUT);
pinMode(LGREEN, OUTPUT);
pinMode(LYELLOW, OUTPUT);

servo.attach(SERVO_PIN);
servo.write(90);

WiFi.softAPConfig(local_IP, gateway, subnet);
WiFi.softAP(ssid, password);
```

```
Serial.begin(57600);
}

void loop() {
digitalWrite(CONVEYOR_MOTOR_PIN, HIGH); // start conveyor

if (digitalRead(IR_SENSOR_PIN) == LOW) {
//Serial.println("Sensor HIGH");
digitalWrite(CONVEYOR_MOTOR_PIN, LOW); // stop conveyor
//Serial.println("Conveyor stop");
//Serial.println("Wait 2 sec");

Serial.println("Connecting...");
digitalWrite(LYELLOW, HIGH);
delay(2000);

WiFiClient client;
const int httpPort = 8080;
if (!client.connect("192.168.0.106", httpPort)) {
  Serial.println("Connection failed");
  return;
}

client.print(String("GET / HTTP/1.1\r\n") +
             "Host: 192.168.0.106\r\n" +
             "Connection: close\r\n\r\n");

while (!client.available()) {
  delay(10);
}

Serial.println("Connected");
digitalWrite(LYELLOW, LOW);
```

```
String line = "";
bool quoteEncountered = false;
char response;
while (client.available()) {
    char ch = client.read();
    Serial.print(ch);
    if (ch == '\'' && !quoteEncountered) {
    quoteEncountered = true;
    } else if (quoteEncountered) {
    response = ch;
    //Serial.println(ch);
    break;
}}
client.stop();

int unk = 0;
Serial.println("received response:");
if (response == '0') {
  Serial.println("Bio");
  digitalWrite(LGREEN, HIGH);
  servo.write(0);
} else if (response == '1') {
  Serial.println("Non-Bio");
  digitalWrite(LRED, HIGH);
  servo.write(180);
} else if (response == '2') {
  Serial.println("Empty");
  unk = 1;
} else {
  Serial.println("Unknown response:");
  Serial.println(response);
  unk = 1;
}
```

```
//Serial.println("Conveyor start");
digitalWrite(CONVEYOR_MOTOR_PIN, HIGH); // start conveyor
//Serial.println("Let the object move (2 sec)");
if(unk==0){
  delay(5000);
}
digitalWrite(LGREEN, LOW);
digitalWrite(LRED, LOW);
} else {
//Serial.println("Sensor LOW");
}
}
```

2. Android Studio Code Snippets

2.1 Snippet from Classifier.java

```
public abstract class Classifier {
  public static String ModelOut;
  MyHttpServer st = new MyHttpServer(8080);
...
}
...
public static class Recognition {
...
public List<Recognition> recognizeImage(final Bitmap bitmap,
int sensorOrientation) {
    // Logs this method so that it can be analyzed with
systrace.
    Trace.beginSection("recognizeImage");

    Trace.beginSection("loadImage");
    long startTimeForLoadImage = SystemClock.uptimeMillis();
    inputImageBuffer = loadImage(bitmap, sensorOrientation);
    long endTimeForLoadImage = SystemClock.uptimeMillis();
```

```
    Trace.endSection();
    LOGGER.v("Timecost to load the image: " +
(endTimeForLoadImage - startTimeForLoadImage));

    // Runs the inference call.
    Trace.beginSection("runInference");
    long startTimeForReference = SystemClock.uptimeMillis();
    tflite.run(inputImageBuffer.getBuffer(),
outputProbabilityBuffer.getBuffer().rewind());
    long endTimeForReference = SystemClock.uptimeMillis();
    Trace.endSection();
    LOGGER.v("Timecost to run model inference: " +
(endTimeForReference - startTimeForReference));

    // Gets the map of label and probability.
    Map<String, Float> labeledProbability =
            new TensorLabel(labels,
probabilityProcessor.process(outputProbabilityBuffer))
                    .getMapWithFloatValue();
    Trace.endSection();



    Log.d("sas", String.valueOf(ModelOut));
    //Log.d("sas", String.valueOf(labeledProbability.get("1
NonBio")));

    try {
      if (labeledProbability.get("0 Bio") > 0.5) {
        ModelOut = "'0' Bio";

      } else if (labeledProbability.get("1 NonBio") > 0.5) {
        ModelOut = "'1' NonBio";

      } else {
```

```
        ModelOut = "'2' Empty";
      }
    } catch (Exception e) {
      System.out.println(e);
    }
    st.setResults(ModelOut);
    try {
      st.start();
    } catch (IOException e) {
      e.printStackTrace();
    }

    return getTopKProbability(labeledProbability);
}
...
}
```

2.2 ClassifierQuantizedMobileNet.java

```
package org.tensorflow.lite.examples.classification.tflite;

import android.app.Activity;
import java.io.IOException;
import org.tensorflow.lite.support.common.TensorOperator;
import org.tensorflow.lite.support.common.ops.NormalizeOp;

/** This TensorFlow Lite classifier works with the quantized
MobileNet model. */
public class ClassifierQuantizedMobileNet extends Classifier {

  /**
   * The quantized model does not require normalization, thus
set mean as 0.0f, and std as 1.0f to
   * bypass the normalization.
```

```
   */
  private static final float IMAGE_MEAN = 0.0f;

  private static final float IMAGE_STD = 1.0f;

  /** Quantized MobileNet requires additional dequantization
to the output probability. */
  private static final float PROBABILITY_MEAN = 0.0f;

  private static final float PROBABILITY_STD = 255.0f;

  /**
   * Initializes a {@code ClassifierQuantizedMobileNet}.
   *
   * @param activity
   */
  public ClassifierQuantizedMobileNet(Activity activity,
Device device, int numThreads)
      throws IOException {
    super(activity, device, numThreads);
  }
  @Override
  protected String getModelPath() {
    return "converted_tflite/model.tflite";
  }
  @Override
  protected String getLabelPath() {
    return "converted_tflite/labels.txt";
  }
  @Override
  protected TensorOperator getPreprocessNormalizeOp() {
    return new NormalizeOp(IMAGE_MEAN, IMAGE_STD);
  }
  @Override
```

```
  protected TensorOperator getPostprocessNormalizeOp() {
    return new NormalizeOp(PROBABILITY_MEAN, PROBABILITY_STD);
  }
}
```

2.3 MyHttpServer.java

```
package org.tensorflow.lite.examples.classification.tflite;

import fi.iki.elonen.NanoHTTPD;

public class MyHttpServer extends NanoHTTPD {
    private String results;

    public MyHttpServer(int port) {
        super(port);
    }

    public void setResults(String results) {
        this.results = results;
        //Log.d("setResults", String.valueOf(results));
    }

    @Override
    public Response serve(IHTTPSession session) {
        //Log.d("serve", String.valueOf(results));
        if (results != null && !results.isEmpty()) {
            return newFixedLengthResponse(results);
        } else {
            String responseText = "No results available";
            return newFixedLengthResponse(responseText);
        }
    }
}
```

REFERENCES

[1] Android Studio (2019). Download Android Studio and SDK tools. [online] Android Studio Available at: https://developer.android.com/studio [Accessed 4 Apr. 2023]

[2] Arduino (2022). Software. [online] www.arduino.cc. Available at: https://www.arduino.cc/en/software

[3] C. P, M.G.,Yadav,S.,Shanmugam, A.,V,H.and Suresh,N.(2021). Waste Classification and Segregation : Machine Learning and IOT Approach. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICIEM51511.2021.944528

[4] Espressif (2020). ESP8266 Technical Reference. [online] Available at: https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf.

[5] Fritzing (2023). Fritzing. [online] fritzing.org. Available at: https://fritzing.org/.

[6] GitHub (2008). GitHub. [online] GitHub. Available at: https://github.com/.

[7] Google LLC (2022). TensorFlow Lite guide. [online] TensorFlow. Available at: https://www.tensorflow.org/lite/guide.

[8] Google LLC (2023). Teachable Machine. [online] teachablemachine.withgoogle.com. Available at: https://teachablemachine.withgoogle.com/ [Accessed 4 Apr. 2023]

[9] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. [online] arXiv.org. Available at: https://arxiv.org/abs/1704.04861.

[10] JGraph Ltd (2023). About diagrams.net. [online] www.diagrams.net. Available at: https://www.diagrams.net/about.

[11] Kang, Z., Yang, J., Li, G. and Zhang, Z. (2020). An Automatic Garbage Classification System Based on Deep Learning. IEEE Access, 8, pp.140019–140029. doi:https://doi.org/10.1109/access.2020.3010496.

[12] NanoHttpd (2023). NanoHttpd/nanohttpd. [online] GitHub. Available at: https://github.com/NanoHttpd/nanohttpd [Accessed 5 Apr. 2023].

[13] Out, N.C. (2008). DUAL FULL-BRIDGE DRIVER. [online] Available at: https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf.

[14] Patel, R. and Chaware, A. (2021). Quantizing MobileNet Models for Classification Problem. [online] IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/9441115 [Accessed 4 Jan. 2023].

[15] Simonyan, K. and Zisserman, A. (2015). Published as a conference paper at ICLR 2015 VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. [online] Available at: https://arxiv.org/pdf/1409.1556.pdf.

[16] TensorFlow (2022). examples/lite/examples/image_classification/android at master · tensorflow/examples. [online] GitHub. Available at: https://github.com/tensorflow/examples/tree/master/lite/examples/image_classification/android [Accessed 4 Apr. 2023].

[17] Texas Instruments (2023). LM2596 SIMPLE SWITCHER ® Power Converter 150-kHz 3-A Step-Down Voltage Regulator. [online] Available at: https://www.ti.com/lit/ds/symlink/lm2596.pdf.