A NOVEL FRAMEWORK FOR AUTOMATIC OBJECT DETECTION AND CLASSIFICATION OF BIO-DEGRADABLE AND NON-BIO-DEGRADABLE WASTE MATERIALS USING DEEP LEARNING

A PROJECT REPORT

Submitted by

AGNES MARY LAVANYA A

BRINDHA G

SAHANA SRI D

VINODHA R

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

MAY, 2025

A NOVEL FRAMEWORK FOR AUTOMATIC OBJECT DETECTION AND CLASSIFICATION OF BIO-DEGRADABLE AND NON-BIO-DEGRADABLE WASTE MATERIALS USING DEEP LEARNING

A PROJECT REPORT

Submitted by

AGNES MARY LAVANYA A (811721243004)

BRINDHA G (811721243012)

SAHANA SRI D (811721243046)

VINODHA R (811721243062)

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

MAY, 2025

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report titled **“A NOVEL FRAMEWORK FOR AUTOMATIC OBJECT DETECTION AND CLASSIFICATION OF BIO-DEGRADABLE AND NON-BIO-DEGRADABLE WASTE MATERIALS USING DEEP LEARNING”** is the bonafide work of **AGNES MARY LAVANYA A (811721243004) BRINDHA G (811721243012) SAHANA SRI D (811721243046) VINODHA R (811721243062),** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr. T.Avudaiappan M.E., Ph.D.,	Mr. A. Joshua Issac M.E.,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Department of Artificial Intelligence K.Ramakrishnan College of Technology (Autonomous) Samayapuram – 621 112	ASSISTANT PROFESSOR Department of Artificial Intelligence K.Ramakrishnan College of Technology (Autonomous) Samayapuram – 621 112

Submitted for the viva-voce examination held on ………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **"A NOVEL FRAMEWORK FOR AUTOMATIC OBJECT DETECTION AND CLASSIFICATION OF BIO-DEGRADABLE AND NON-BIO-DEGRADABLE WASTE MATERIALS USING DEEP LEARNING"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfilment of the requirement of the award of Degree of **BACHELOR OF TECHNOLOGY**.

Signature

____

AGNES MARY LAVANYA A

____

BRINDHA G

____

SAHANA SRI D

____

VINODHA R

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and in-debt to our institution "**K. Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

We are glad to credit honourable chairman **Dr. K.RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

We would like to thank **Dr. N. VASUDEVAN, M.E., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

We whole heartily thank to **Dr. T.AVUDAIAPPAN**, **M.E., Ph.D.,** Head of the department, **ARTIFICIAL INTELLIGENCE** for providing his encourage pursuing this project.

We express our deep and sincere gratitude to our project guide **Mr. A. JOSHUA ISSAC**, **M.E.**, Department of **ARTIFICIAL INTELLIGENCE,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

We render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

We wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

ABSTRACT

Waste management is a challenge for modern cities as it impacts on environmental sustainability and on the level of quality perceived by citizens. The Digital Transformation process in this sector will require time other than a careful management of the waste segregation associated with the waste collection processes. At this regard, the progress of hardware and software technologies enabling the Internet of Things (IoT) will contribute significantly to accelerate the whole process. Among these innovative solutions Artificial Intelligence (AI), the traditional waste management system can be replaced with smart sensors embedded into the system to perform real time monitoring and allow for better waste management. The project is to develop a smart waste management system using IoT communication protocol and AI based predictive deep learning model. Hardware sensors sense the bin status data and Image Processing program performs real time object detection and classification. Object detection and waste classification is done in AI based framework with pre-trained object detection model. This object detection model is trained with images of waste to generate a frozen inference graph used for object detection which is done through a camera connected to the PC as the main processing unit. Ultrasonic sensor is embedded into each waste compartment to monitor the filling level of the waste. The intelligent smart bin, application, visualization and decision-making platform into a system, which is the most, complete and effective system among the known research works. The results of the test we conducted on our platform using our extended dataset showed that our scheme is very reliable.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

GSM	-	GLOBAL SYSTEM FOR MOBILE COMMUNICATION
IDE	-	INTEGRATED DEVELOPMENT ENVIRONMENT
LCD	-	LIQUID CRYSTAL DISPLAY
LED	-	LIGHT EMITTING DIODE
PCB	-	PRINTED CIRCUIT BLOCK
USB	-	UNIVERSAL SERIAL BUS

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

The current process of waste management starts with the waste being created by people in the cities and disposed in trash bins near its creation point. The disposed trash is collected by municipality or private company trucks at the predefined times and transferred to temporary collection centres. The trash at the collection centres is then sent for recycling. Some of these problems can be mitigated by implementing smart waste management systems.

This process in current city setting solves the waste problem partially while it creates other problems such as;

- Some trash bins are overfilled while others are under filled by the trash collection time,
- overfilled trash bins create unhygienic conditions,
- Un-optimized truck routes result in excessive fuel usage and environmental pollution and
- All collected trash is combined which complicates sorting at the recycling facility.

1.2 PROBLEM STATEMENT

The waste management services take care of a healthy environment allowing optimization of the utilities and prevent overloading the carrier for waste disposal. Smart waste management also contributes to the overall waste recycling efficiency and provides the route optimization opportunity for utilities to reduce traffic and fuel use. An example of a modern smart waste management system would include; a sensor attached to the trash bin that measures fill level; and a communication system that transfers this data to Cloud. Data is processed in the cloud, thus, the route of collection trucks is optimized. Smart waste management companies have recently developed solutions based on ultrasonic distance measurement. Some companies prefer to approach the problem with an alternative solution using image processing and camera as a passive sensor. However, the majority of these solutions use ultrasonic sensor for measurement of the distance. Ultrasonic sensors use a well-known sonar technique to perform measurement of the distance between the sensor and an obstacle. The sensor consists of an emitter that sends the sound pulse and a receiver that detects that pulse upon its reflection from an object. The distance to an object can be determined by measuring the time between emitting and receiving a pulse since the speed of sound is known. As stated in Table 1, besides the determination of the fill level by implementing different kinds of sensors additional data may be obtained. These sensors monitor other parameters related to the status of the container such as position using GPS or motion sensors to register container collections, movement to detect possible vandalism.

1.3 OBJECTIVE

The first is related to the usage of the ultrasonic sensor. Trash is non-uniformly distributed inside the container. Simple distance measurement leads to false fill level measurement. Although several software procedures were proposed to increase the accuracy of this sensor, unfortunately, results remain poor. By using multiple sensors, the fill level can be determined more precisely, however, the cost of the system also increases. So, this is usually not a commercially preferred solution.The waste management cycle starts with the garbage being produced, then it gets disposed at the local trash bins or other garbage collection points. Afterward, it is being collected by a garbage collecting company and brought to garbage depot where it is being sorted and sent for recycling, destruction (burning), or storage. Complete waste management should be involved in trash bin fill level measurement, route optimization of the trucks, and contribution to the recycling process by easing out the sorting process, which is currently manual and slow. Current smart waste management services do not offer any solution with regard to the recycling process.

CHAPTER 2

LITERATURE SURVEY

2.1 EDGE OF THINGS: THE BIG PICTURE ON THE INTEGRATION OF EDGE, IOT AND THE CLOUD IN A DISTRIBUTED COMPUTING ENVIRONMENT

H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty, C.T

The writers expound on the development of these technologies as central elements in contemporary computational platforms, highlighting their complementary nature in data treatment, storage, and analysis. Edge computing handles data near the source, hence minimizing latency and bandwidth consumption, whereas cloud computing provides large capacity for storage and processing capacity. By combining the two technologies, IoT devices can run more efficiently by offloading the heavy computations to the cloud while facilitating real-time processing at the edge. By using case studies and practical applications, the authors demonstrate successful instances where this convergence has led to better data management structures and operational efficiencies.

Merits: Offers a comprehensive view of technology integration and practical applications in enhancing system efficiencies.

Demerits: Lacks in-depth analysis of specific algorithms and their performance metrics.

2.2 A COLLABORATIVE INTERNET OF THINGS ARCHITECTURE FOR SMART CITIES AND ENVIRONMENTAL MONITORING

Montori, F., Bedogni, L., Bononi, L

This article suggests a new collaborative Internet of Things (IoT) architecture specifically for smart cities, with a main emphasis on environmental monitoring. The authors contend that urban management using conventional methods tends to be inadequate in effectively handling the complex relationship between different urban systems. Thus, the need for a collaborative system taking advantage of collective intelligence among IoT devices is critical. Core elements of this architecture are device interoperability, data-sharing protocols, and enhanced analytics capabilities that enable collaborative decision-making among city stakeholders. The paper also discusses potential challenges, such as data privacy issues, scalability of systems, and standardised protocols. Real-life case studies are discussed, illustrating how this collaborative architecture can be successfully utilized in cities to aid monitoring campaigns as well as encourage sustainability measures. Finally, the paper urges more research into collaborative approaches that can ensure intelligent urban living while responding to serious environmental issues.

Merits: Promotes resource efficiency and sustainability through collaborative approaches in IoT deployment.

Demerits: Implementation challenges in real-world scenarios are not thoroughly addressed.

2.3 SMART AND GREEN URBAN SOLID WASTE COLLECTION SYSTEMS: ADVANCES, CHALLENGES, AND PERSPECTIVES

J. W. Lu, N. B. Chang, L. Liao, M. Y. Liao

The research examines the development of urban solid waste collection systems with a focus on incorporating smart technologies to develop more effective and ecologically sound strategies. The authors point out the growing challenges that cities are experiencing with respect to waste management, such as growing waste production, dwindling landfill space, and the resultant environmental consequences of poor waste disposal strategies. Through an examination of current technologies and approaches, the paper outlines how intelligent waste management systems can take advantage of IoT devices, sensor networks, and data analytics to streamline waste collection routes, track bin levels in real time, and improve service responsiveness. Notable developments explored are the implementation of smart bins with sensors and mobile applications for real-time data transmission, which enable proactive decision-making. The authors conclude with future directions in research and recommend interdisciplinary efforts that bring together technology, urban planning, and community participation to establish sustainable waste management systems.

Merits: Highlights the importance of smart solutions in waste management and their environmental impact.

Demerits: Offers limited practical examples of successful implementations.

2.4 CHALLENGES AND OPPORTUNITIES OF WASTE MANAGEMENT IN IOT-ENABLED SMART CITIES: A SURVEY

T. Anagnostopoulos, A. Zaslavsky, K. Kolomvatsos, A. Medvedev, P. Amirian, J. Morley, S. Hadjieftymiades

This survey surveys the revolutionary role of IoT technology in waste management in the background of smart cities. Through syntheses of prevailing literature, authors present major hurdles and possible challenges arising from adoption of IoT-based systems in waste management in the city. The article classifies the challenges into various areas of focus, such as technical challenges such as data interoperability, network stability, and the exorbitant costs of installing IoT infrastructures. On the opportunity aspect, the writers explain how Internet of Things-enabled solutions, such as fill-level sensors in intelligent bins and location-tracking waste management trucks, will result in strong operational efficiencies. To solve challenges addressed, authors invoke further studies using standardized protocols and strengthened security implementations to create integrated partnerships between many stakeholders within cities involved in the management of their wastes.

Merits: Provides a broad overview of current challenges and innovative solutions within the domain.

Demerits: Generalized survey findings may overlook localized issues and specific case studies.

2.5 INCORPORATING INTELLIGENCE IN FOG COMPUTING FOR BIG DATA ANALYSIS IN SMART CITIES

B. Tang, Z. Chen, G. Hefferman, S. Pei, T. Wei, H. He, Q. Yang

This article presents the incorporation of intelligent processing functionalities in fog computing platforms to support big data analysis for smart city application. As urban spaces increasingly produce huge amounts of data through various sensors and devices, the shortcomings of conventional cloud-based computing architectures manifest in terms of latency, bandwidth consumption, and real-time computational capabilities. The authors suggest a hybrid model that integrates fog computing and advanced analytics so that data can be processed near its source. This minimizes the data sent to the cloud, thereby reducing traffic on the cloud network and enhancing response times. The paper discusses several algorithms employed for data filtering, aggregation, and real-time analytics in the context of fog computing, though particular algorithms are mentioned in broad terms and not in detail. Through case studies highlighting successful implementation, the paper demonstrates the real-world applications of intelligent fog computing in different areas of smart city, including transport, energy, and public safety.

Merits: Enhances processing speed and efficiency for big data analytics in urban environments.

Demerits: The complexity of implementing fog computing solutions in existing infrastructures is not deeply explored.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

For an IoT-based solution to be implemented, it should be energy efficient, able to communicate, and share information across extended coverage. An IoT-based embedded system is with GSM communication technology is used as the platform to perform data transmission to the server. Web-based Android applications are developed to interface with a web server to provide information from sensors monitoring bin status, amount of waste in the bin, and time of waste collection. The data are processed by a graph theory optimization algorithm to obtain the shortest path for reaching the bin to efficiently manage the waste collection strategies.

A second IoT-based smart bin comes with three compartments, each with its own functionality: The first compartment consists of an infrared IR sensor and metal detector. The second compartment consists of an IR sensor and moisture sensor to detect dry and wet waste. The last compartment is subdivided into three bins for the collection of segregated waste respectively. The system connects to WiFi for data transmission to a specific server. The storage compartment consists of a rotating table with three bins, namely dry, wet, and metal. It rotates according to the type of waste detected in the previous compartment. The use of Wi-Fi as a communication medium limits its transmission range, which is a crucial element of a smart bin that might be situated in a remote location.

A third IoT-based solid waste management system is a DHT22 temperature sensor, MQ-135 gas sensor, IR sensor, passive infrared, PIR sensor, and load cell are used to monitor the temperature and humidity, presence of harmful gas, amount of garbage, presence of user, and weight of garbage respectively. LoRa communication is used to transmit data to a gateway, and the data are sent to a cloud for cloud monitoring. The system uses a total of five waste bins to handle five different types of wastes, with each bin having its own set of sensors, which ultimately increases the overall cost of the system.

A fourth IoT-based system relies on an ultrasonic sensor to monitor the amount of waste in the bin. The monitored data are also transmitted through LoRa communication. The proposed system only monitors the amount of waste in the bin. However, the author highlighted it power management with such components as a counter and switching regulator to manage the power consumption

In the existing method the ultrasonic sensor and motion sensor is used. Automatic lid opening and closing method is used in smart dustbin and microcontroller programming is done in the system. Since smart cities are becoming center of attraction for the advancement of developing countries and without the removal or solution to the garbage problem these cities will be not that attractive. Therefore, large number of projects and research is going on in the area of smart dustbins for smart cities and to implement such projects typically use microcontroller based real time bin monitoring system, RFID technology, GPS, GSM, RF module etc.

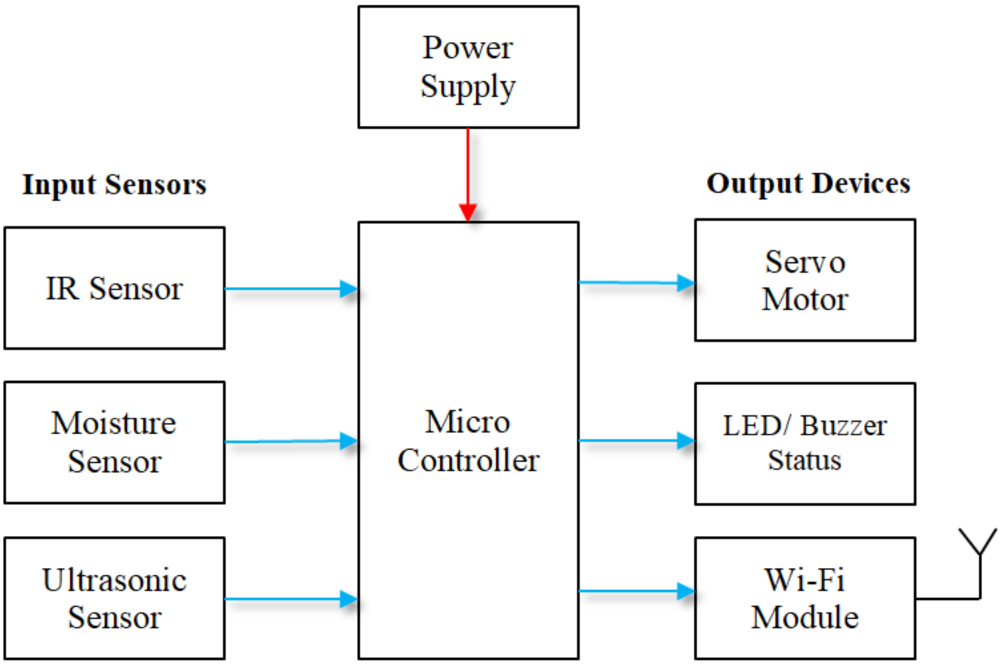

Fig. 3.1 Existing System Model

3.1.1 Drawbacks

- Manual operation makes less accurate
- Make more pollutant and poisoning gases.
- It creates unhygienic conditions for people as well as ugliness to that place leaving bad smell.

3.2 PROPOSED SYSTEM

Conventional bins are usually categorized in terms of the type of waste, for example, recyclable and non-recyclable waste. The recyclable bin is further categorized into different types, such as paper, metal, and plastic waste. This convention has resulted in as many as 4 different types of bin situated at a garbage collection point. This eventually increases the overall cost of operation for the maintenance of the bin. Even if the designated bins are well prepared for public usage, often the public will not utilize it properly and simply throw waste in any of the bins regardless of designation. Hence, conventional bins have proved their ineffectiveness in the public eyes. This project offers a solution to this issue by having separated waste compartments account for different types of waste, such as plastic and general waste. In order to effectively identify and segregate different types of waste, an object detection model is trained using a Tensorflow framework and exported to a Python Language to perform waste detection. Ultrasonic sensor monitors the filling level of the bin, while a GPS module monitors its location. The status of the bin's filling level and location is then sent to the server through an IoT module for the purpose of monitoring.

- An AI predictive system is proposed, where the system evaluates the condition of equipment using predictive maintenance techniques.
- Moreover, the bin itself is scattered around the city where connectivity to the database might not be feasible.
- If we were to reduce the latency of waste classification, PC connectivity would be chosen to upload the image at a higher data rate.
- This would imply that the bin must be within that range in order to classify waste, which is not ideal.

- Hence, the system would perform better by performing waste classification on the board itself to reduce the latency of waste classification.

3.2.1 Advantages

- In this system, a 24x7 monitoring system is designed for monitoring dumpsters.
- Here a smart and organized system is designed for selective clearing.
- The ultrasonic sensor is used for measuring the level of waste in the dumpster.
- DC motor powered platform is used for segregating wet and dry waste.

CHAPTER 4

SYSTEM SPECIFICATIONS

4.1 HARDWARE SPECIFICATIONS

- Arduino NANO
- Power supply
- LCD
- Ultrasonic Sensor
- IR Sensor
- Servo Motor
- USB Interface

4.2 SOFTWARE SPECIFICATIONS

- Arduino IDE
- Python IDE
- Proteus 7 Professional

4.3 HARDWARE DESCRIPTION

Power Supplies

A power supply (sometimes known as a power supply unit or PSU) is a device or system that supplies electrical or other types of energy to an output load or group of loads. The term is most commonly applied to electrical energy supplies, less often to mechanical ones, and rarely to others.

This circuit is a small +5V power supply, which is useful when experimenting with digital electronics. Small inexpensive wall transformers with variable output voltage are available from any electronics shop and supermarket. Those transformers are easily available, but usually their voltage regulation is very poor, which makes then not very usable for digital circuit experimenter unless a better regulation can be achieved in some way. The following circuit is the answer to the problem.

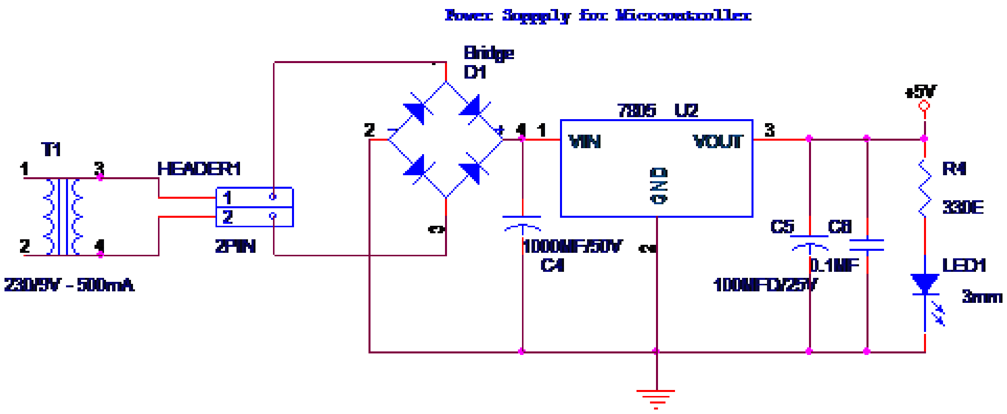

Fig. 4.1 Block Diagram of Power Supply

Transformer

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled wires. A changing current in the first circuit (the primary) creates a changing magnetic field; in turn, this magnetic field induces a changing voltage in the second circuit (the secondary). By adding a load to the secondary circuit, one can make current flow in the transformer, thus transferring energy from one circuit to the other. The secondary induced voltage V_S is scaled from the primary V_P by a factor ideally equal to the ratio of the number of turns of wire in their respective windings:

$$\frac{V_S}{V_P} = \frac{N_S}{N_P}$$

By appropriate selection of the numbers of turns, a transformer thus allows an alternating voltage to be stepped up — by making N_S more than N_P or stepped down, by making it less.

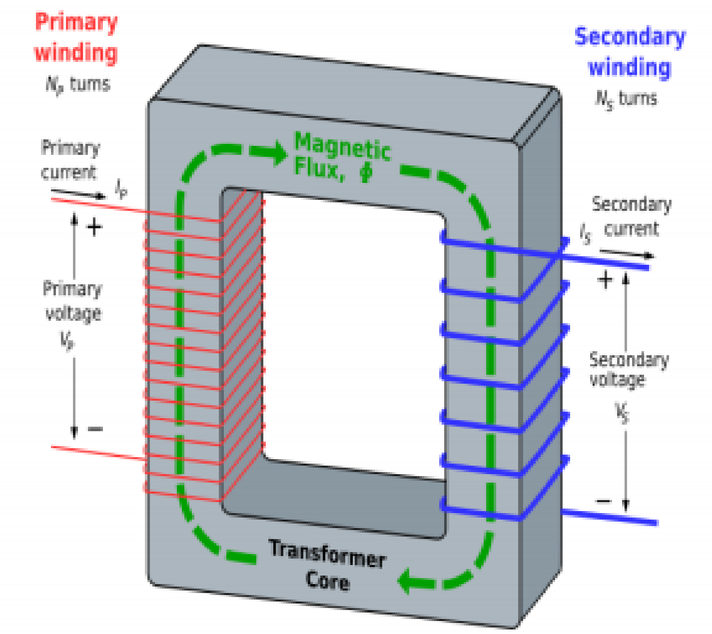

Fig.4.2 An Ideal Step-Down Transformer

A simplified an ideal step-down transformer design is shown in the above figure. A current passing through the primary coil creates a magnetic field. The primary and secondary coils are wrapped around a core of very high magnetic permeability, such as iron; this ensures that most of the magnetic field lines produced by the primary current are within the iron and pass through the secondary coil as well as the primary coil.

Rectifier

A rectifier is an electrical device that converts alternating current (AC), which periodically reverses direction, to direct current (DC), which flows in only one direction. The process is known as rectification. Rectifiers are used as components of power supplies and as detectors of radio signals. Mainly there are three types of rectifier i.e. half wave rectifier, full wave rectifier and Bridge Rectifier.

Half-Wave Rectifier

In half-wave rectification of a single-phase supply, either the positive or negative half of the AC wave is passed, while the other half is blocked. Because only one half of the input waveform reaches the output, mean voltage is lower. Half-wave rectification requires a single diode in a single-phase supply, or three in a three-phase supply. Rectifiers yield a unidirectional but pulsating direct current; half-wave rectifiers produce far more ripple than full-wave rectifiers, and much more filtering is needed to eliminate harmonics of the AC frequency from the output.

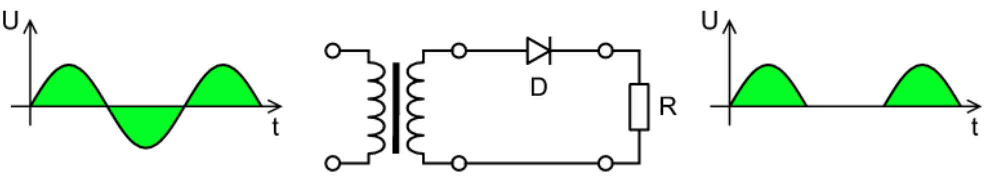

Fig. 4.3 Half Wave Rectifier

Full-Wave Rectifier

A full-wave rectifier converts the whole of the input waveform to one of constant polarity (positive or negative) at its output. Full-wave rectification converts both polarities of the input waveform to pulsating DC (direct current), and yields a higher average output voltage. Two diodes and a centre tapped transformer are needed.

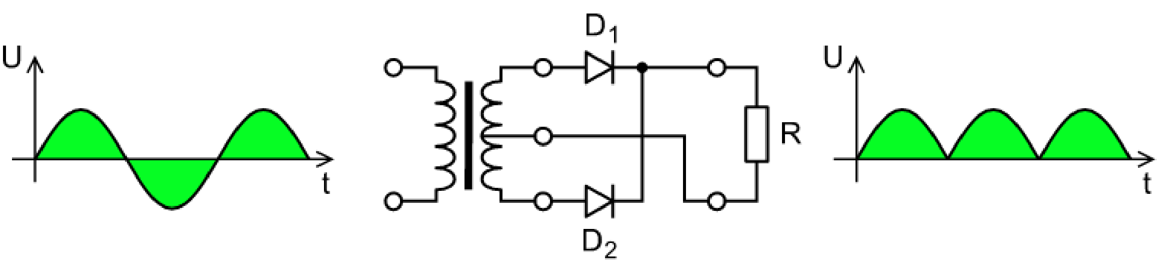

Fig. 4.4 Full-Wave Rectifier

Bridge Rectifier

A diode bridge is an arrangement of four (or more) diodes in a bridge circuit configuration that provides the same polarity of output for either polarity of input. When used in its most common application, for conversion of an alternating current (AC) input into a direct current (DC) output, it is known as a bridge rectifier. A bridge rectifier provides full-wave rectification from a two-wire AC input, resulting in lower cost and weight as compared to a rectifier with a 3-wire input from a transformer with a centre-tapped secondary winding. The essential feature of a diode bridge is that the polarity of the output is the same regardless of the polarity at the input.

Basic operation

According to the conventional model of current flow, current is defined to be positive when it flows through electrical conductors from the positive to the negative pole. In actuality, free electrons in a conductor nearly always flow from the negative to the positive pole. In the vast majority of applications, however, the actual direction of current flow is irrelevant. Therefore, in the discussion below the conventional model is retained.

In the diagrams below, when the input connected to the left corner of the diamond is positive, and the input connected to the right corner is negative, current flows from the upper supply terminal to the right along the red (positive) path to the output, and returns to the lower supply terminal via the blue (negative) path.

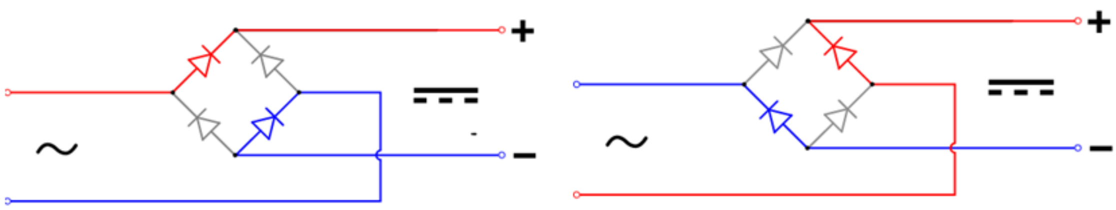

Fig. 4.5 Operation of Bridge Rectifier

IC Voltage Regulators

Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection all in a single IC. Although the internal construction of the IC is somewhat different from that described for discrete voltage regulator circuits, the external operation is much the same. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustable set voltage. A power supply can be built using a transformer connected to the ac supply line to step the ac voltage to desired amplitude, then rectifying that ac voltage, filtering with a capacitor and RC filter, if desired, and finally regulating the dc voltage using an IC regulator. The regulators can be selected for operation with load currents from hundreds of milliamperes to tens of amperes, corresponding to power ratings from milliwatts to tens of watts.

Three-Terminal Voltage Regulators

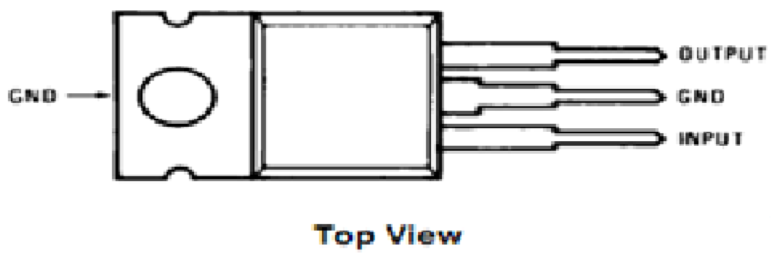

Fig.4.6 Three-Terminal Voltage Regulators

Figure shows the basic connection of a three-terminal voltage regulator IC to a load. The fixed voltage regulator has an unregulated dc input voltage, V_i, applied to one input terminal, a regulated output dc voltage, V_o, from a second terminal, with the third terminal connected to ground. For a selected regulator, IC device specifications list a voltage range over which the input voltage can vary to maintain a regulated output voltage over a range of load current. The specifications also list the amount of output voltage change resulting from a change in load current (load regulation) or in input voltage (line regulation). The series 78 regulators provide fixed regulated voltages from 5 to 24 V. Figure shows how one such IC, a 7805, is connected to provide voltage regulation with output from this unit of +5V dc. An unregulated input voltage V is filtered by capacitor C1 and connected to the IC's IN terminal. The IC's OUT terminal provides a regulated + 12V which is filtered by capacitor C2 (mostly for any high-frequency noise). The third IC terminal is connected to ground (GND). While the input voltage may vary over some permissible voltage range, and the output load may vary over some acceptable range, the output voltage remains constant within specified voltage variation limits. These limitations are spelled out in the manufacturer's specification sheets. There are two types of voltage regulator they are 78xx series and 79xx series.

78xx series

There are common configurations for 78xx ICs, including 7805 (5 V), 7806 (6 V), 7808 (8 V), 7809 (9 V), 7810 (10 V), 7812 (12 V), 7815 (15 V), 7818 (18 V), and 7824 (24 V) versions. The 7805 is the most common, as its regulated 5-volt supply provides a convenient power source for most TTL components. Less common are lower-power versions such as the LM78Mxx series (500 mA) and LM78Lxx series (100 mA) from National Semiconductor. Some devices provide slightly different voltages than usual, such as the LM78L62 (6.2 volts) and LM78L82 (8.2 volts) as well as the STMicroelectronics L78L33ACZ (3.3 volts).

79xx series

The 79xx devices have a similar "part number" to "voltage output" scheme, but their outputs are negative voltage, for example 7905 is -5 V and 7912 is -12 V. The 7912 has been a popular component in ATX power supplies, and 7905 was popular component in ATX before -5 V was removed from the ATX specification.

Table 4.1 Positive Voltage Regulators in 7800 series

IC Part	Output Voltage (V)	Vi (V)
7805	+5	7.3
7806	+6	8.3
7808	+8	10.5
7810	+10	12.5
7812	+12	13.6
7815	+15	17.7
7818	+18	21.0
7824	+24	27.1

Arduino Nano

Arduino Nano has similar functionalities as Arduino Duemilanove but with a different package. The Nano is inbuilt with the ATmega328P microcontroller, same as the Arduino UNO. The main difference between them is that the UNO board is presented in PDIP (Plastic Dual-In-line Package) form with 30 pins and Nano is available in TQFP (plastic quad flat pack) with 32 pins. The extra 2 pins of Arduino Nano serve for the ADC functionalities, while UNO has 6 ADC ports but Nano has 8 ADC ports. The Nano board doesn't have a DC power jack as other Arduino boards, but instead has a mini-USB port. This port is used for both programming and serial monitoring. The fascinating feature in Nano is that it will choose the strongest power source with its potential difference, and the power source selecting jumper is invalid.

Arduino Nano Pinout Description

Taking this pin-out diagram below as reference, we shall discuss all the functionalities of each and every pin.

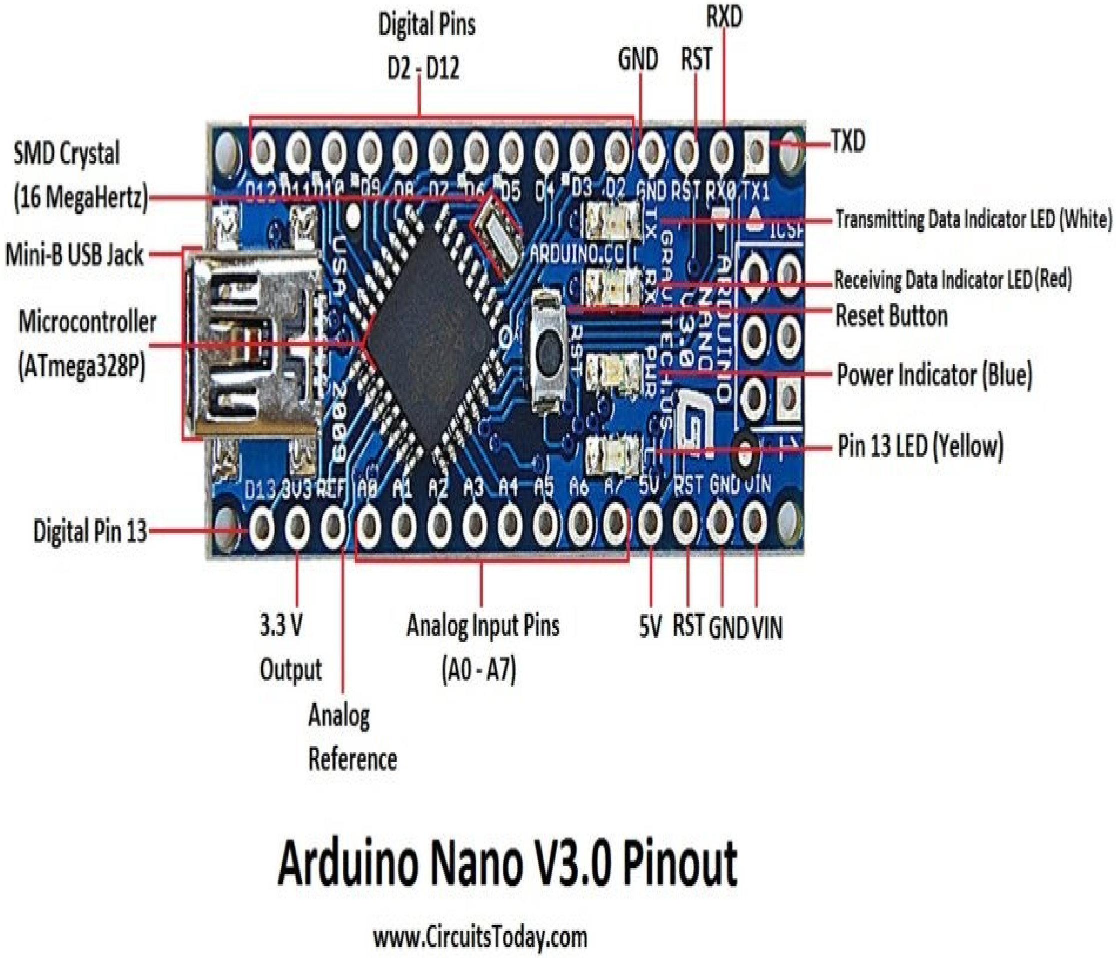

Fig.4.7 Arduino Nano V3.0 Pinout

We can infer from the image that Arduino Nano got 36 pins in total. We will see all the pins section wise as well as a detailed format at last.

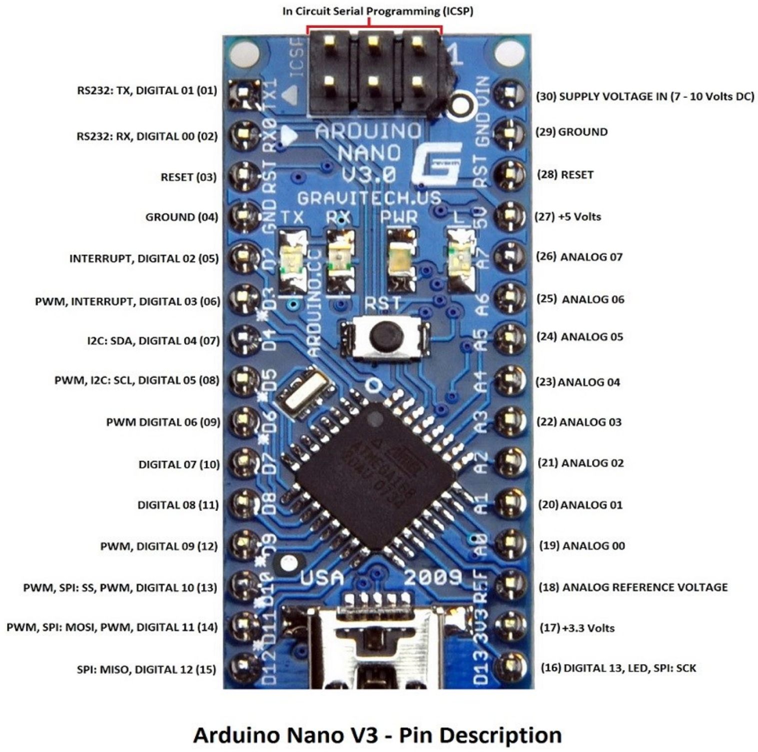

Fig 4.8 Arduino Nano V3 – Pin Description

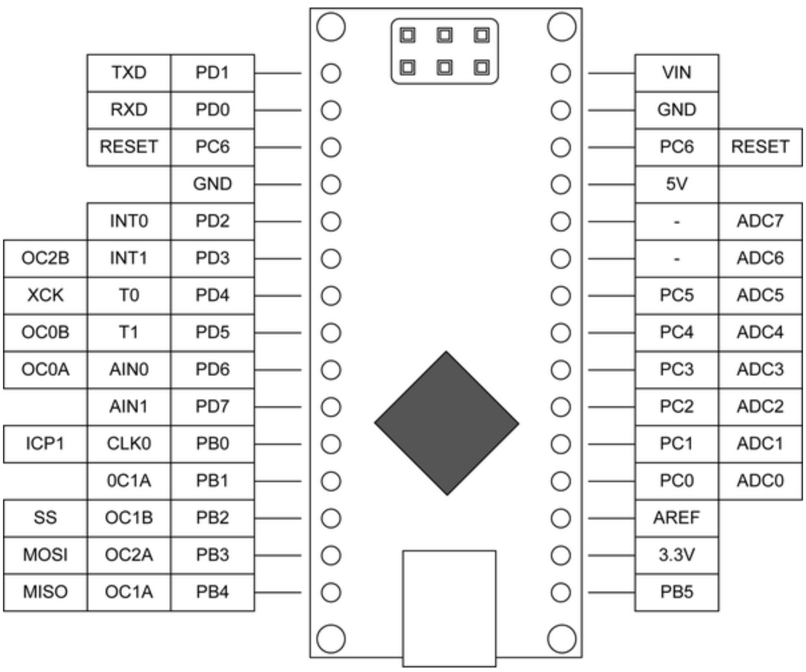

Fig .4.9 Arduino Nano V3 – Pin

Memory

Memory can be broadly divided into 3 classes:

- 32KB Flash memory –This is the storage space of the compiled program of which the boot loader uses 0.5 KB.
- 2KB SRAM – This is mainly used during run time.
- 1KB EEPROM –This is used for storing data that should not be erased upon switching off power.

Power Setup

The Arduino R3 operates at 5 Volts. It can either be powered through USB cable from the computer or through the DC jack provided on the Board.

The DC Jack

The voltage regulator 7805 is provided in the board for obtaining 5v regulated output voltage. The input voltage applied can be between 7-25 volts DC power.

USB Power

When powered through the USB, the 500mA Re-settable fuse on the USB power line is used to abstain the board from drawing current in excess.

USB Connectivity

Since the ATmega328 does not use USB communication directly, the need for a dedicated IC arises. FTDI FT232 IC is used to communicate between the microcontroller and USB serially. The drivers required for the Serial to USB converter has to be installed.

Hardware

The Arduino R3 / Arduino Uno Boards have 20 programmable I/O's. They are grouped mainly as

- Pins 0 to 13
- Pins 0 to 5 [Analog Inputs 0 to 5]
- Pins 0 to 5 [Analog Inputs 0 to 5]

Digital I/O's

The 20 I/O’s can accept digital signals as input as well as outputs. The digital pins are numbered from 0 to 19. The Digital Pins can be used for controlling LED's, Relays and for accepting input from Push-Buttons, Digital Sensors

Analog I/O's

Analog inputs can be given to pins A0-A5. A inbuilt ADC analog to digital converter is present that converts analog voltages in the range of 0 to 5 volts to a 10-bit value. Analog sensors that sense changes in temperature or light can work with these inputs.

Analog Output

The six pins marked PWM are pins dedicated to produce Analog Output Signal. They can produce analog voltages in the range of 0 to 5 volts with a resolution of 8-bits. They can be used for Intensity Control, Speed Control, Etc.

LCD (Liquid Crystal Display)

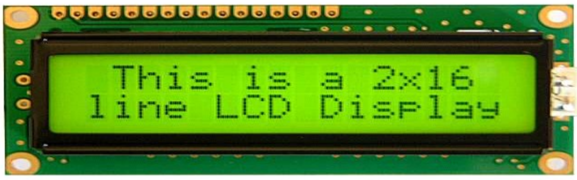

Fig. 4.10 LCD Display

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD, the data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Liquid crystal displays are used for display of numeric and alphanumeric character in dot matrix and segmental displays. The two liquid crystal materials which are commonly used in display technology are nematic and cholesteric whose schematic arrangement of molecules.

Based on the construction, LCD's are classified into two types. They are,

(i) Dynamic scattering type

(ii) Field effect type.

Dynamic Scattering Type

The construction of the dynamic scattering liquid crystal cell is shown in the fig. The display consists of two glass plates, each coated with tin oxide (SnO2) on the inside with transparent electrodes separated by a liquid crystal layer,5µA to 50 µA thick. The oxide coating on the front sheet is etched to produce a single or multi- segment pattern of characters, with each segment properly insulated from each other. A weak electric field applied to liquid crystal tends to align molecule in the direction of the field. As soon as the voltage exceeds certain threshold value, the domain structure collapses and the appearance is changed. As the voltage grows further, the flow becomes turbulent and the substance turns optically homogenous. In this disordered state, the liquid crystal scatters light.

Field Effect Type

The construction of the field effect LCD display is similar to that of the dynamic scattering type, with the expectation that two thin polarizing optical filters are placed at the inside of each glass sheet. The LCD material is of twisted nematic type which twists the light (change in direction of polarization) passing through the cell when the latter is not energized. This allows light to pass through the optical filters and the cell appears bright. When the cell is energized, no twisting of light takes place and the cell appears dull. Liquid crystals consume small amount of energy, in a seven segment display the current drawn is about 25 µA for dynamic scattering cells and 300 µA for field effect cells LCD's require ac voltage supply. A typical voltage supply to dynamic scattering LCD's are normally used for seven-segmental displays.

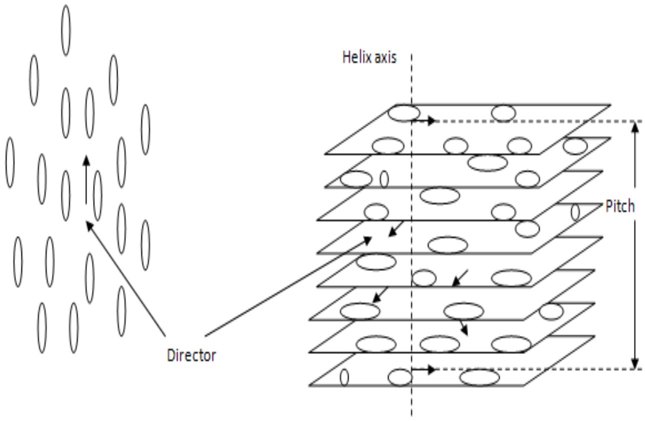

Fig. 4.11 Schematic Arrangement in Liquid Crystal

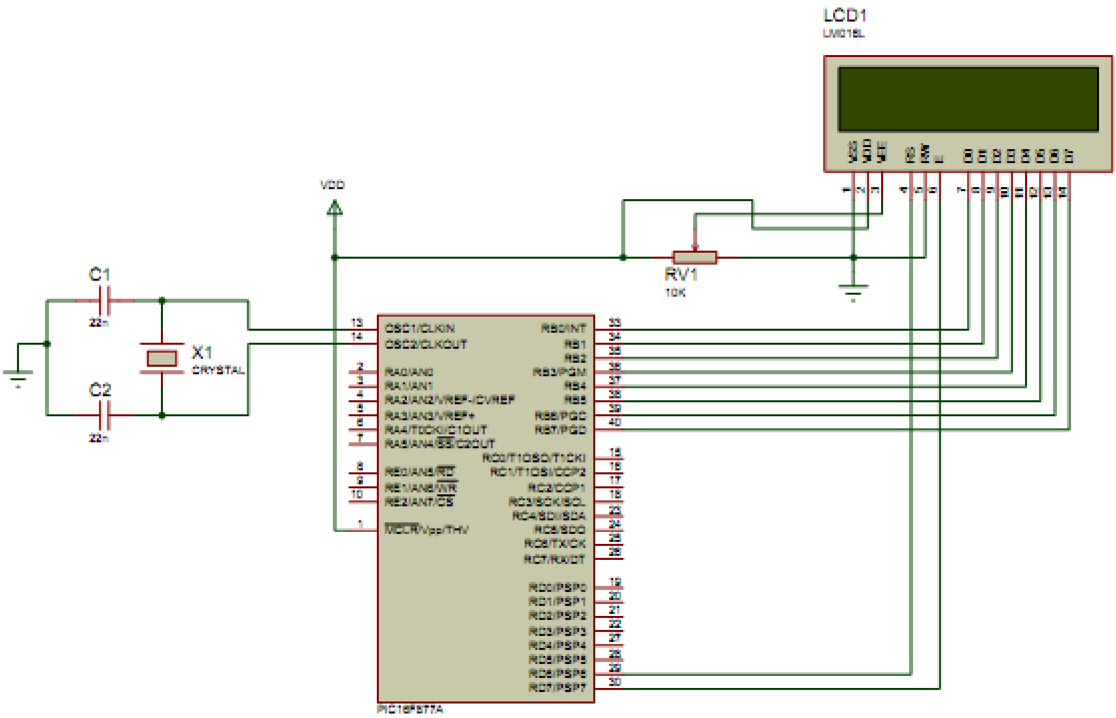

Fig.4.12 LCD Interface with PIC16F8

4.4 SOFTWARE DESCRIPTION

Embedded C

An embedded system is an application that contains at least one programmable computer (typically in the form of a microcontroller, a microprocessor or digital signal processor chip) and which is used by individuals who are, in the main, unaware that the system is computer-based.

Embedded Systems Programming

Embedded systems programming is different from developing applications on desktop computers. Key characteristics of an embedded system, when compared to PCs, are as follows. Embedded devices have resource constraints (limited ROM, limited RAM, limited stack space, less processing power) Components used in embedded system and PCs are different; embedded systems typically use smaller, less power consuming components. Embedded systems are more tied to the hardware. Two salient features of Embedded Programming are code speed and code size. Code speed is governed by the processing power, timing constraints, whereas code size is governed by available program memory and use of programming language. Goal of embedded system programming is to get maximum features in minimum space and minimum time.

Embedded systems are programmed using different type of language

- Machine Code
- Low level language, i.e., assembly
- High level language like C, C++, Java, Ada, etc.
- Application-level language like Visual Basic, scripts, Access, etc.

Keil C51 C Compilers

- Direct C51 to generate a listing file
- Define manifest constants on the command line
- Control the amount of information included in the object file
- Specify the level of optimization to use
- Specify the memory models

Proteus

Proteus PCB design electronic circuits can computer-aided design and circuit boards are designed.

ISIS (Intelligent Schematic Input System)

ISIS includes a base VSM engine with support for the following functions:

- DC / AC voltmeter and ammeter, oscilloscopes, logic analysers
- Analog signal generators, digital pattern generator
- Timer functions, protocol analysers (including RS232, I2C, SPI)

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

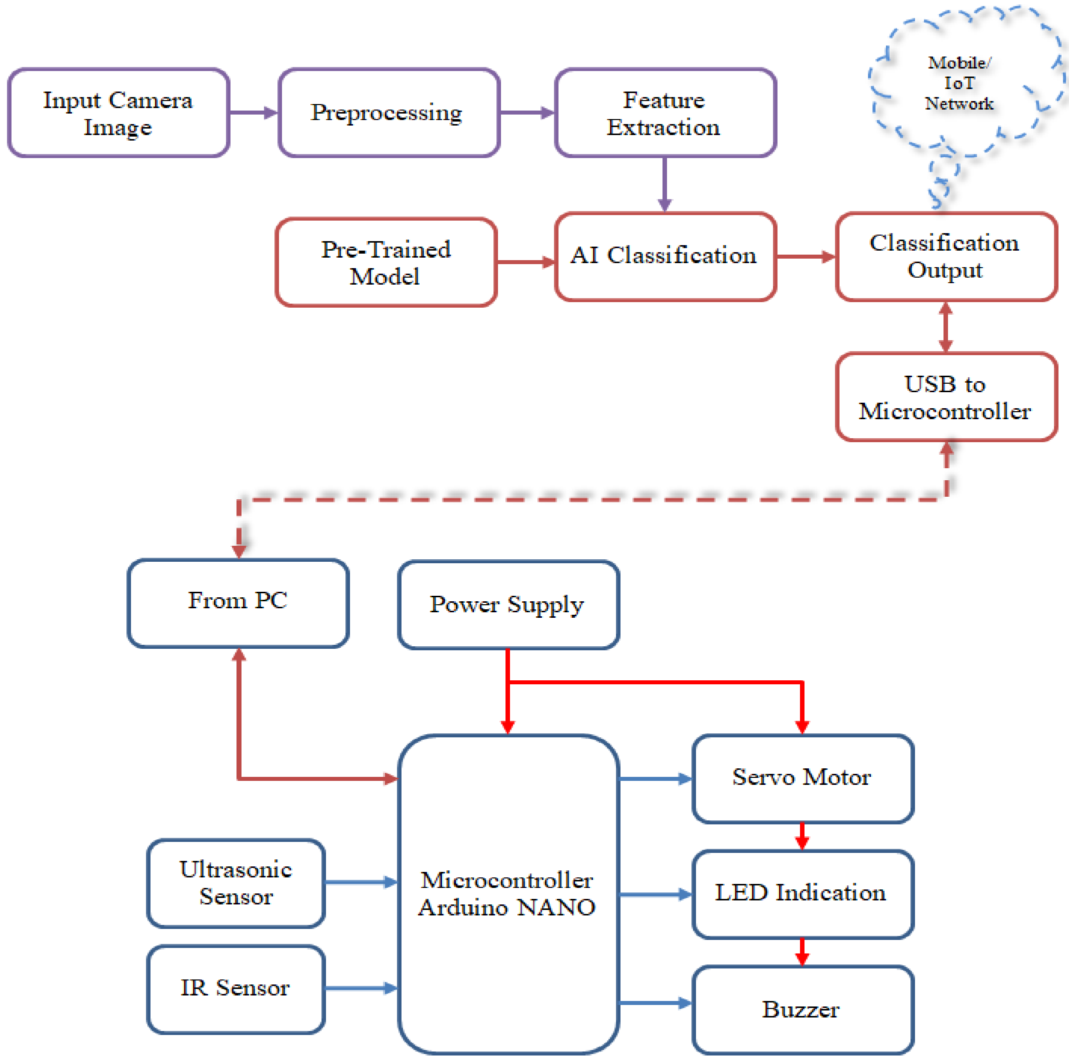

Fig.5.1 System Architecture

This block diagram represents of the bin using the modeling. The electronic compartment holds the electronic components. The waste detection compartment has a retractable platform that holds the waste temporarily. At the same time, waste identification is being performed by capturing the image of the waste and processing it with the PC. The bin is designed with two different compartments to hold plastic waste and general waste.

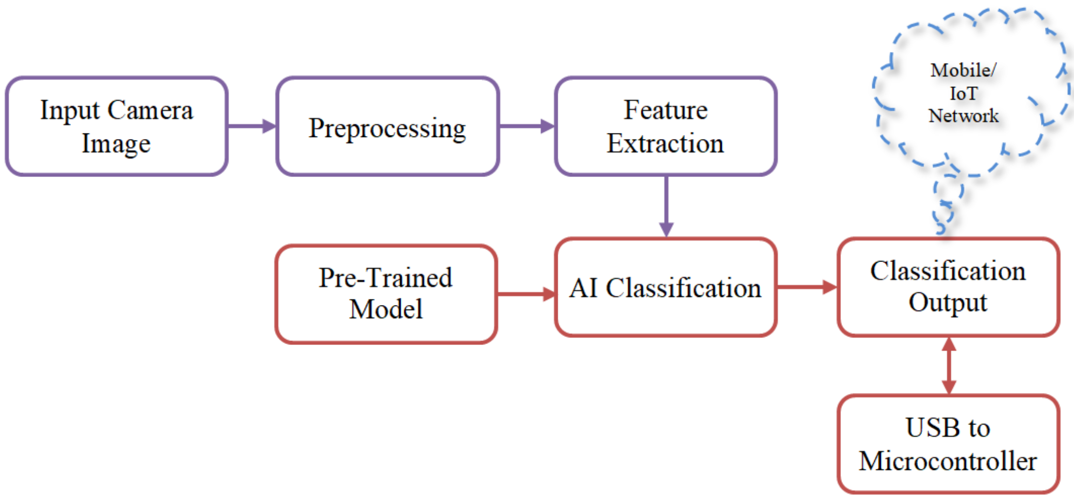

Fig.5.2 Statistical Analysis Model

A Statistical Analysis System is proposed to analyse and process the data, which requires high computational power. In our proposed system, we have decided to utilize the mobility of Raspberry Pi, a mobile CPU together with MobileNetV2, a lightweight model and a mobile architecture to perform waste classification on the board itself instead of uploading it to the database for cloud analysis. This allows us to reduce the latency in waste classification.

- In this system, a 24x7 monitoring system is designed for monitoring dumpsters.
- Here a smart and organized system is designed for selective clearing.
- The ultrasonic sensor is used for measuring the level of waste in the dumpster.
- DC motor powered platform is used for segregating wet and dry waste.
- If either of the containers is full then an alert message is sent from the dumpster.
- In turn, employees can clear the corresponding dumpster.
- All these sensors are connected to an Arduino Uno board.

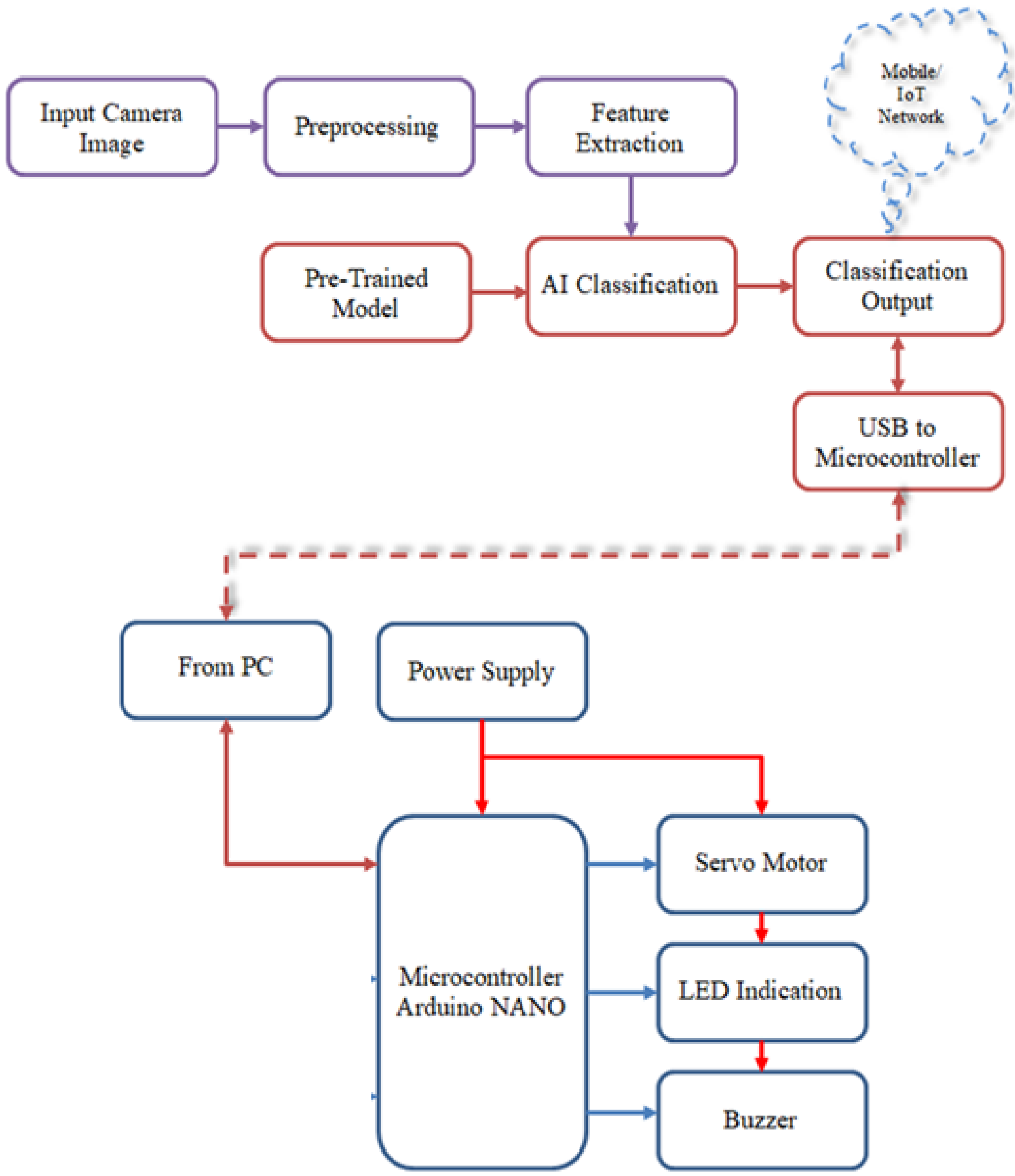

Fig.5.3 Ultrasonic Sensor Model

The ultrasonic sensor is connected to Arduino to monitor the filling level of each of the bin's waste compartment, including a plastic and general waste compartment. The ultrasonic sensor uses sonar to measure the time taken for the signal to travel from the transmitter end to the receiver end, and the time difference is used to calculate the filling level of waste inside the bin. A GPS module provides information on the location (latitude, longitude) as well as the real-time of the bin from the satellite. The filling level, location, and real-time bin are collected and transferred via a IoT module from the bin to the gateway, which is connected to the computer.

- The bin consists of several compartments to segregate the waste including metal, plastic, and paper waste compartment which are controlled by the servo motors.
- Object detection and waste classification is done in AI framework with pre-trained object detection model.

5.2 DATA FLOW DIAGRAM

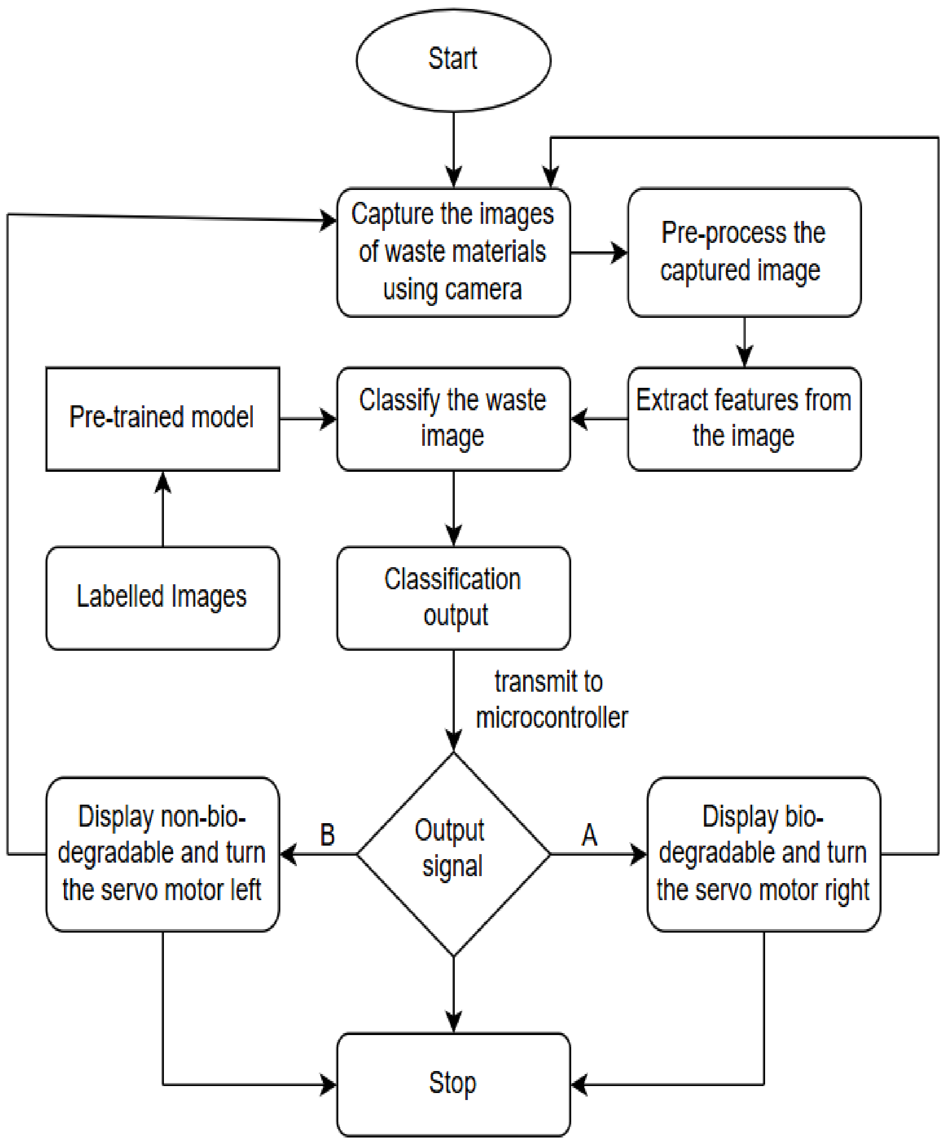

Fig.5.4 Data Flow Diagram

5.3 USE CASE DIAGRAM

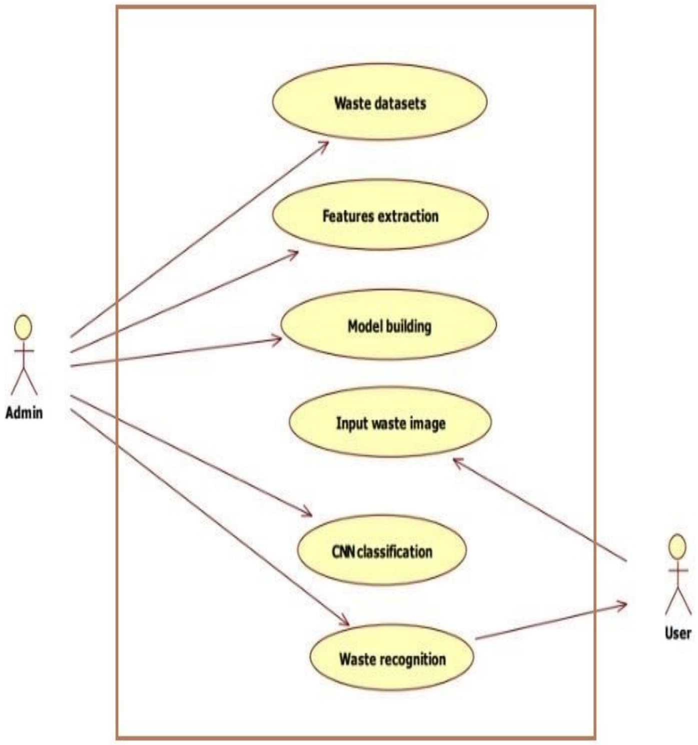

Fig.5.5 Use Case Diagram

5.4 ACTIVITY DIAGRAM

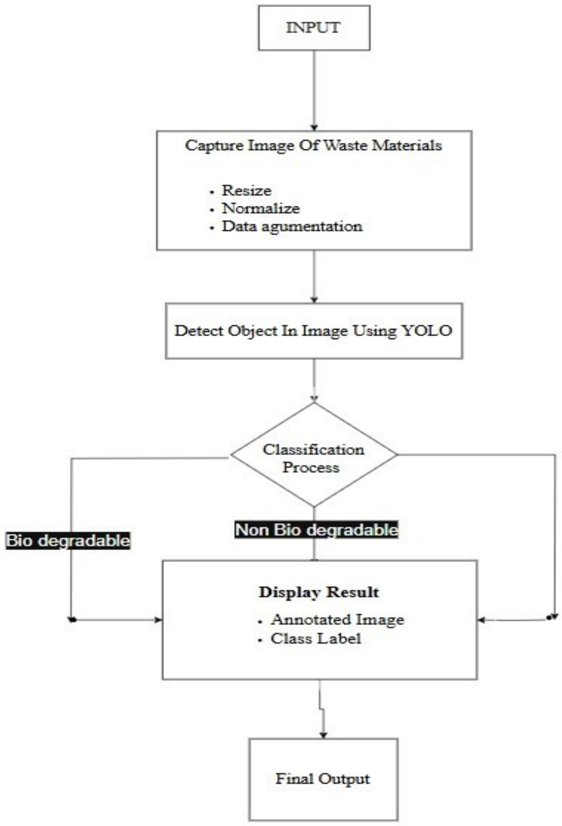

Fig.5.6 Activity Diagram

CHAPTER 6

MODULES DESCRIPTION

6.1 LIST OF MODULES

- Sensor Implementation Module
- Object Detection Module
- Waste Classification Module

6.2 SENSORS IMPLEMENTATION MODULE

The overall block diagram represents, where the development process of the smart bin is given. Arduino Uno and Python based image processing system operate independently and communicate with each other between USB interface cables. The camera module is connected to PC to capture the waste image for the purpose of object detection and identification.

After the waste is identified, servo motors controlled by the Arduino will actuate the opening and closing of the lid of the waste compartment. The opening of the lid allows waste to fall from the waste detection compartment into its respective waste compartment.

The ultrasonic sensor is connected to Arduino NANO to monitor the filling level of each of the bin's waste compartment, including a plastic, metal waste compartment. The ultrasonic sensor uses sonar to measure the time taken for the signal to travel from the transmitter end to the receiver end, and the time difference is used to calculate the filling level of waste inside the bin. A Sensor node id module provides information on the location (latitude, longitude) as well as the real-time of the bin from the satellite. The filling level, location, and real-time bin are collected and transferred via a WiFi module from the bin to the IoT gateway, which is connected to the computer.

6.3 OBJECT DETECTION MODULE

A predictive system proposed, where the system evaluates the condition of equipment using predictive maintenance techniques.

The data input of video or image is handling in PC python program also Arduinoused to pre-process the data collected from the sensors, and the data is uploaded to the database through USB interface connectivity for cloud analysis.

A Statistical Analysis System is proposed to analyze and process the data, which requires high computational power. In our proposed system, we have decided to utilize the mobility of PC, a CPU together with MobileNetV2, a lightweight model and a mobile architecture to perform waste classification on the board itself instead of uploading it to the database for cloud analysis. This allows us to reduce the latency in waste classification. Moreover, the bin itself is scattered around the city where connectivity to the database might not be feasible. For example, a 5MP image has a typical file size of 15.0MB. If we were to reduce the latency of waste classification, Wi-Fi connectivity would be chosen to upload the image at a higher data rate. However, Wi-Fi connectivity is limited by its transmission range of around 50m. This would imply that the bin must be within that range in order to classify waste, which is not ideal. Hence, the system would perform better by performing waste classification on the board itself to reduce the latency of waste classification.

6.4 WASTE CLASSIFICATION MODULE

Waste identification is performed using the TensorFlow object detection API running on the Raspberry Pi. This object detection API runs on a pre-trained object detection model, SSD MobileNetV2, which is lightweight and suitable to run on low-computing power devices such as python programming. The architecture of MobileNetV2 is based on linear bottlenecks depth-separable convolution with inverted residuals and it is an improvement over the previous version, MobileNetV1. Depth-separable convolution requires less computation by splitting convolution into two separate layers, depth wise convolution and point wise convolution. Figure 6 represents the operation of depth-wise convolution and pointwise convolution. Depth-wise convolution is performed by extracting spatial features of each input feature separately, thereby reducing the number of parameters and computational cost. On the other hand, pointwise convolution is a 1×1 convolution used to build new features through linear combination from the output of depth-wise convolution.

6.5 HARDWARE DEVELOPING ENVIRONMENT

Arduino software (ide)

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

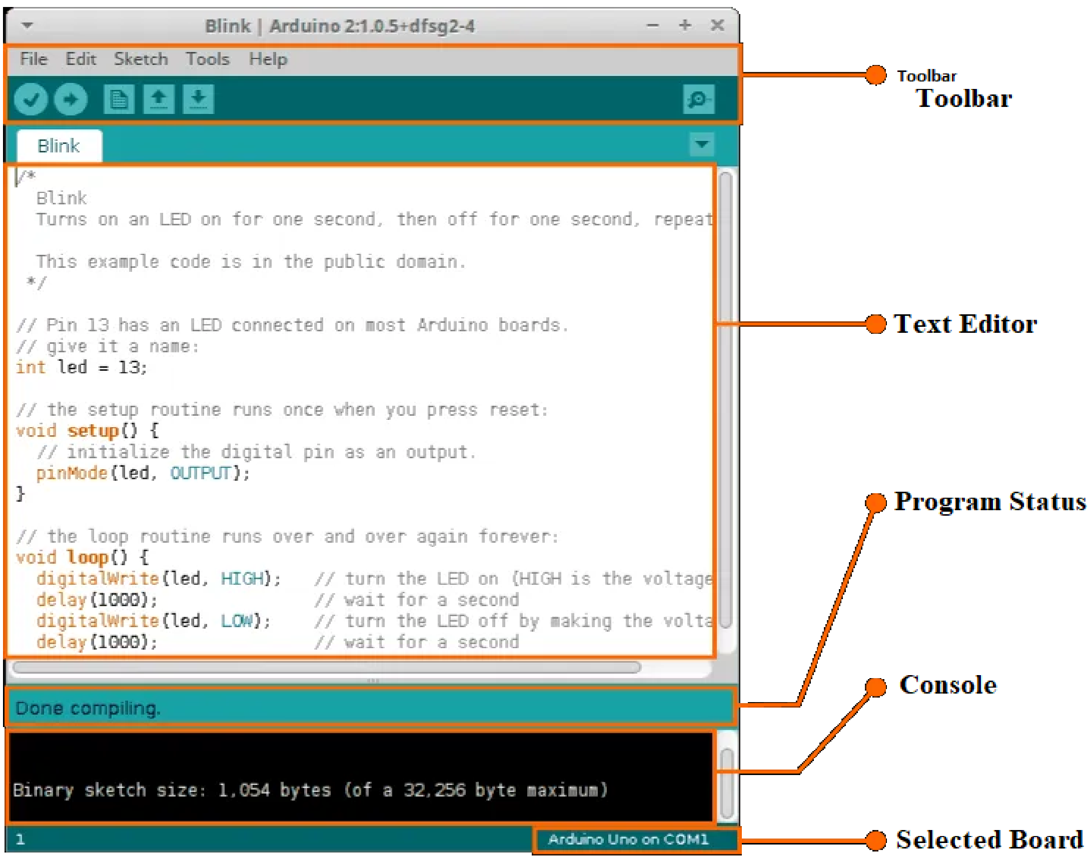

Fig.6.1 Arduino IDE

Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension ".ino". The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

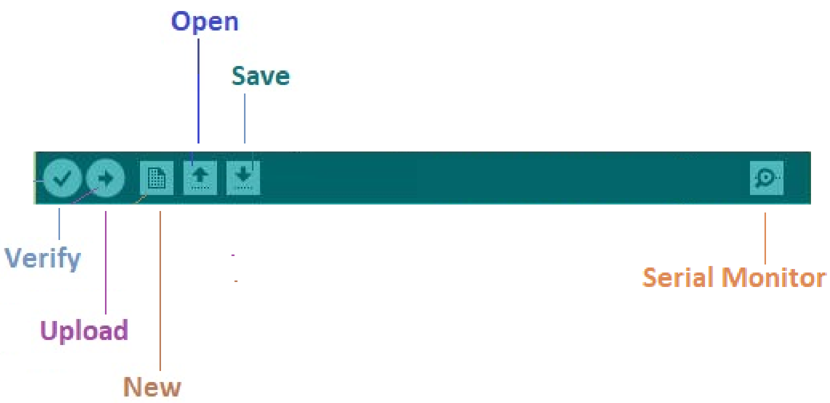

Fig.6.2 Arduino IDE Tool Icon

- **Verify** Checks your code for errors compiling it.
- **Upload** Compiles your code and uploads it to the configured board. See uploading below for details. Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"
- **New** Creates a new sketch.
- **Open** Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.
- **Save** Saves your sketch.
- **SerialMonitor** Opens the serial monitor.

Additional commands are found within the five menus: File, Edit, Sketch, Tools, and Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

File

- **New** Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.
- **Open** Allows to load a sketch file browsing through the computer drives and folders.
- **Open** Recent Provides a short list of the most recent sketches, ready to be opened.
- **Sketchbook** Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.
- **Examples** Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.
- **Close** Closes the instance of the Arduino Software from which it is clicked.
- **Save** Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.
- **Save as...** Allows to save the current sketch with a different name.
- **Page Setup** It shows the Page Setup window for printing.
- **Print** Sends the current sketch to the printer according to the settings defined in Page Setup.
- **Preferences** Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.
- **Quit** Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.

Edit

- **Undo/Redo** Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.
- **Cut** Removes the selected text from the editor and places it into the clipboard.
- **Copy** Duplicates the selected text in the editor and places it into the clipboard.
- **Copy for Forum** Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.
- **Copy as HTML** Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.
- **Paste** Puts the contents of the clipboard at the cursor position, in the editor.
- **Select All** Selects and highlights the whole content of the editor.
- **Comment/Uncomment** Puts or removes the // comment marker at the beginning of each selected line.
- **Increase/Decrease** Indent Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.
- **Find Opens** the Find and Replace window where you can specify text to search inside the current sketch according to several options.
- **Find Next** Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.
- **Find Previous** Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

Sketch

- **Verify/Compile** Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.
- **Upload** Compiles and loads the binary file onto the configured board through the configured Port.
- **Upload Using Programmer** This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload

to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools -> Burn Bootloader command must be executed.

- **Export** Compiled Binary Saves a .hex file that may be kept as archive or sent to the board using other tools.
- **Show Sketch Folder** Opens the current sketch folder.
- **Include Library** Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.
- Add File... Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right sideo the toolbar.

Tools

- **Auto Format** This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.
- **Archive Sketch** Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.
- **Fix Encoding & Reload** Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.
- **Serial Monitor** Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.
- **Board** Select the board that you're using. See below for descriptions of the various boards.
- **Port** This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.

- **Programmer** For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this.
- **Burn Bootloader** The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino board but is useful if you purchase a new ATmega microcontroller (which normally comes without a bootloader). Ensure that you've selected the correct board from the Boards menu before burning the bootloader on the target board. This command also set the right fuses.

Help

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to **Getting Started, Reference**, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

Find in Reference This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

Sketchbook

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog.

- Tabs, Multiple Files, and Compilation

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

Before compiling the sketch, all the normal Arduino code files of the sketch (.ino, .pde) are concatenated into a single file following the order the tabs are shown in. The other file types are left as is.

Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx , /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #include statements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

Third-Party Hardware

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

Serial Monitor

This displays serial sent from the Arduino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to Serial.begin in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor.

Preferences

Some preferences can be set in the preferences dialog (found under the Arduino menu on the Mac, or File on Windows and Linux). The rest can be found in the preferences file, whose location is shown in the preference dialog.

Boards

The board selection has two effects: it sets the parameters (e.g. CPU speed and baud rate) used when compiling and uploading sketches; and sets and the file and fuse settings used by the burn bootloader command. Some of the board definitions differ only in the latter, so even if you've been uploading successfully with a particular selection you'll want to check it before burning the bootloader.

Arduino Software (IDE) includes the built-in support for the boards in the following list, all based on the AVR Core. The Boards Manager included in the standard installation allows to add support for the growing number of new boards based on different cores like Arduino Due, Arduino Zero, Edison, Galileo and so on.

Started with the Arduino Nano

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328P. It offers the same connectivity and specs of the UNO board in a smaller form factor.

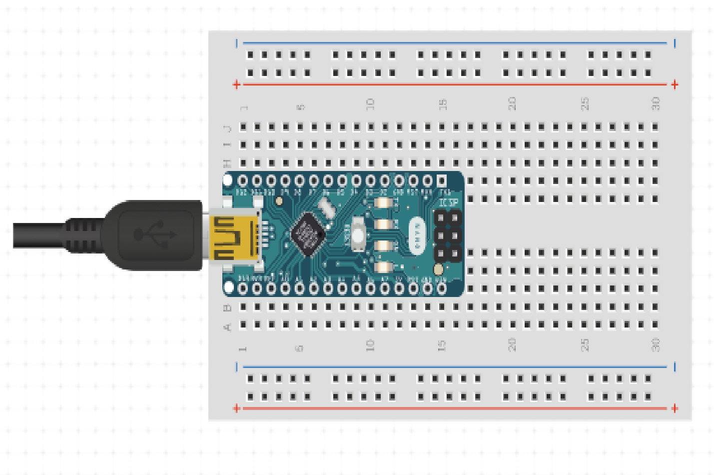

Fig.6.3 Arduino NANO Interface

The Arduino Nano is programmed using the Arduino Software (IDE), our Integrated Development Environment common to all our boards and running both online and offline. For more information on how to get started with the Arduino Software visit the Getting Started page.

Arduino Nano on the Arduino Desktop IDE

If you want to program your Arduino Nano while offline you need to install the Arduino Desktop IDE To connect the Arduino Nano to your computer, you'll need a Mini-B USB cable. This also provides power to the board, as indicated by the blue LED (which is on the bottom of the Arduino Nano 2.x and the top of the Arduino Nano 3.0).

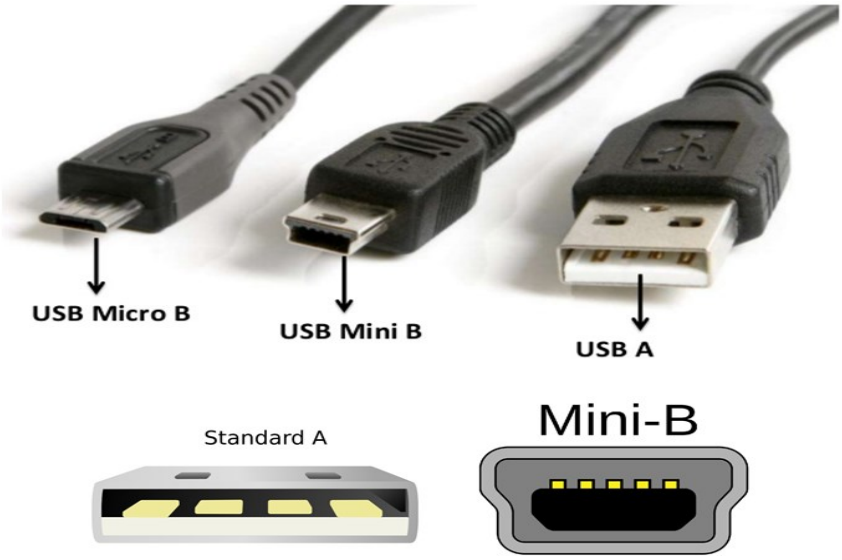

Fig.6.4 NANO Interfacing USB Types Of Port

Open your first sketch

- Open the LED blink example sketch: File > Examples >01.Basics> Blink.

Select your board type and port

- Select Tools > Board > Arduino AVR Boards > Arduino Nano.

NOTE: We have updated the Nano board with a fresh bootloader. Boards sold by us from January 2018 have this new bootloader, while boards manufactured before that date have the old bootloader. First, check that **Tools > Board > Boards Manager**

shows you have the Arduino AVR Boards 1.16.21 or later installed. Then, to program the NEW Arduino NANO boards you need to choose **Tools > Processor > ATmega328P**. To program old boards you need to choose **Tools > Processor > ATmega328P (Old Bootloader)**. If you get an error while uploading or you are not sure which bootloader you have, try each Tools > Processor menu option until your board gets properly programmed.

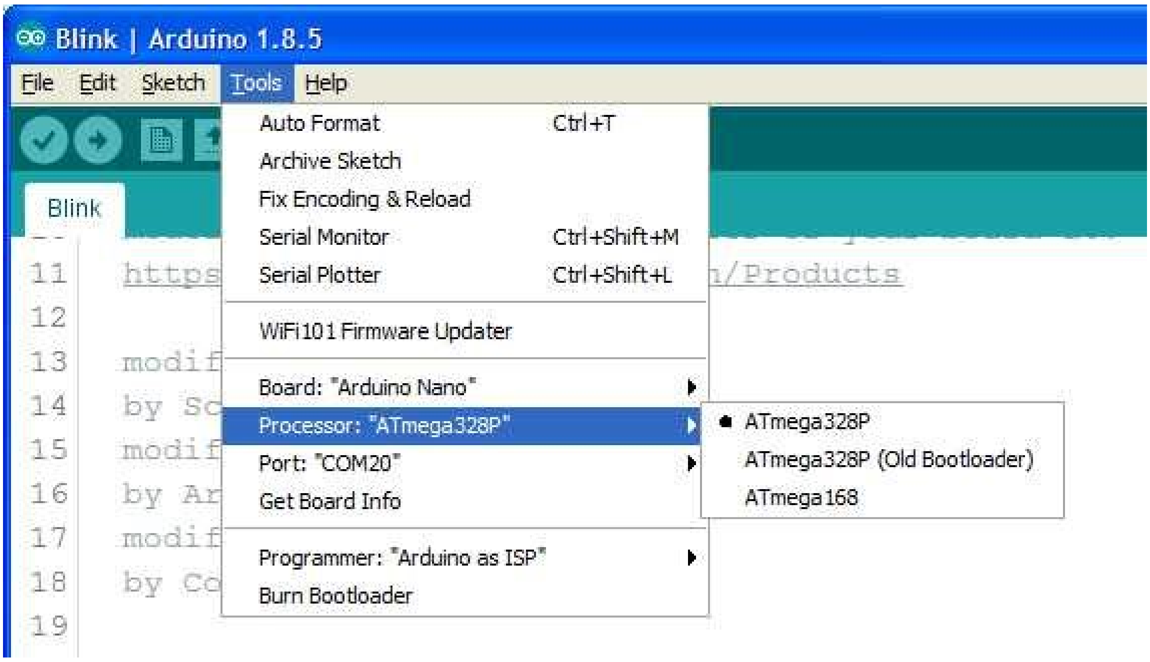

Fig.6.5 Select the NANO Processor Type

Select the serial device of the board from the Tools | Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

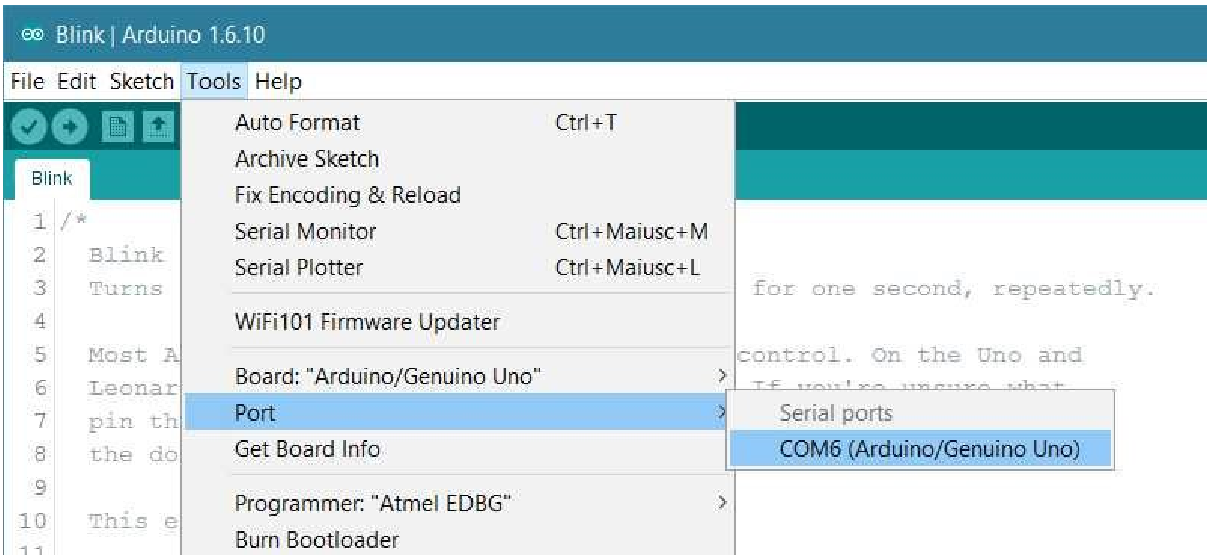

Fig.6.6 Select Board Type

Upload and Run your first Sketch

To upload the sketch to the Arduino Nano, click the Upload button in the upper left to load and run the sketch on your board:

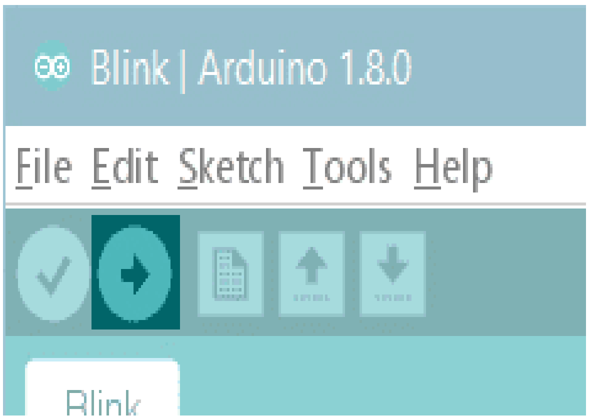

Fig.6.7 Upload to Nano

Wait a few seconds - you should see the RX and TX LEDs on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.

CHAPTER 7

CONCLUSIONS AND FUTURE ENHANCEMENTS

7.1 CONCLUSION

The model was able to detect and classify waste according to classes such as metal, plastic, and paper. However, the accuracy of the model can be improved by increasing the number of training data—in this case, the number of waste images—and by increasing the training time. The segregation of waste is interfaced and coordinated well between the object detection performed by Python program and the servo motor controlling the lid of the individual waste compartment.

7.2 FUTURE ENHANCEMENTS

In the future, the waste detection model is to be improved by increasing the number of waste images in the dataset to increase the flexibility of the system in identifying waste. Moreover, an automated routing system can be developed to identify and pinpoint the shortest path to the bin for the purpose of maintenance.

APPENDIX A

SOURCE CODE

WasteClassification.c

```
#include <Servo.h>


// Pin definitions
const int trigPin = 2;  // Trigger pin for ultrasonic sensor
const int echoPin = 3;       // Echo pin for ultrasonic sensor
const int irSensorPin = 4;   // Input pin for IR sensor
const int ledPin = 9;        // Pin for LED
const int buzzerPin = 10;    // Pin for buzzer
const int servoPin = 6;      // Pin for servo motor


Servo myServo;


void setup() {
Serial.begin(9600);       // Start serial communication
myServo.attach(servoPin); // Attach the servo to the pin
pinMode(ledPin, OUTPUT);  // Set LED pin as output
pinMode(buzzerPin, OUTPUT);// Set Buzzer pin as output
pinMode(trigPin, OUTPUT);  // Set trigger pin as output
```

```
pinMode(echoPin, INPUT);   // Set echo pin as input
pinMode(irSensorPin, INPUT); // Set IR sensor pin as input
}


void loop() {
  if (Serial.available() > 0) {
    int classification = Serial.read() - '0'; // Read the classification from PC


    // Take action based on the classification result
    if (classification == 1) {
myServo.write(90); // Move to a specific position
digitalWrite(ledPin, HIGH); // Turn on LED
tone(buzzerPin, 1000); // Activate buzzer
    } else if (classification == 2) {
myServo.write(0); // Move to another position
digitalWrite(ledPin, LOW); // Turn off LED
noTone(buzzerPin); // Deactivate buzzer
    } else {
myServo.write(0); // Default position
digitalWrite(ledPin, LOW); // Turn off LED
noTone(buzzerPin); // Deactivate buzzer
    }
```

```
    }

    // Ultrasonic sensor measurement
    long duration, distance;

digitalWrite(trigPin, LOW);  // Set trigger pin low
delayMicroseconds(2);
digitalWrite(trigPin, HIGH); // Set trigger pin high
delayMicroseconds(10);        // Keep high for 10 microseconds
digitalWrite(trigPin, LOW);  // Set trigger pin low

    duration = pulseIn(echoPin, HIGH); // Read the echo pin
    distance = (duration * 0.034) / 2; // Convert to distance in cm

    // Check IR sensor
    if (digitalRead(irSensorPin) == HIGH) {
        // If an object is detected by the IR sensor
myServo.write(180); // Move servo to another position
digitalWrite(ledPin, HIGH); // Turn on LED
tone(buzzerPin, 750); // Activate buzzer
delay(1000); // Delay for 1 second
    } else {
```

```
    // No object detected by IR
myServo.write(0); // Move servo to default position
digitalWrite(ledPin, LOW); // Turn off LED
noTone(buzzerPin); // Deactivate buzzer
  }

delay(100); // Delay for stability
}
```

WasteClassification.py

```
import cv2
import numpy as np
import tensorflow as tf

# Load pre-trained model
model = tf.keras.models.load_model('path_to_your_model.h5')

def process_image(image_path):
  # Load and preprocess image
img = cv2.imread(image_path)
img = cv2.resize(img, (224, 224))  # Resize to fit the model input size
img = img.astype('float32') / 255  # Normalize
```

```
img = np.expand_dims(img, axis=0)  # Add batch dimension
    return img


def classify_image(image_path):
img = process_image(image_path)
    prediction = model.predict(img)
class_index = np.argmax(prediction)  # Get the index of the highest probability
    return class_index


# Assuming you have a way to get the image from the camera
image_path = 'path_to_your_captured_image.jpg'
class_index = classify_image(image_path)


# Send classification result to Arduino via USB
import serial
arduino = serial.Serial('COM3', 9600)  # Adjust 'COM3' as needed
arduino.write(str(class_index).encode())
```

APPENDIX B

SCREENSHOTS

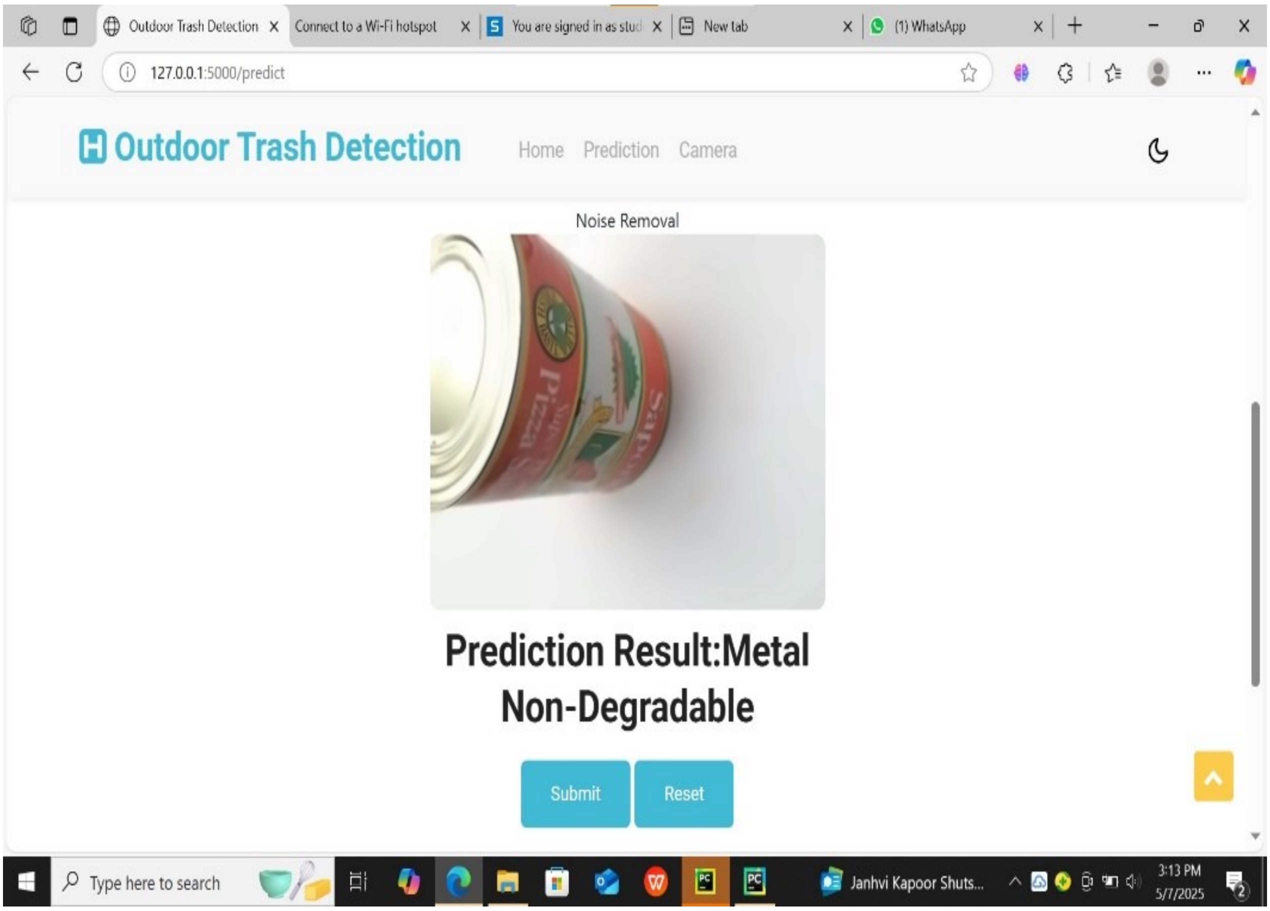

Fig.B.1 Prediction Result 1

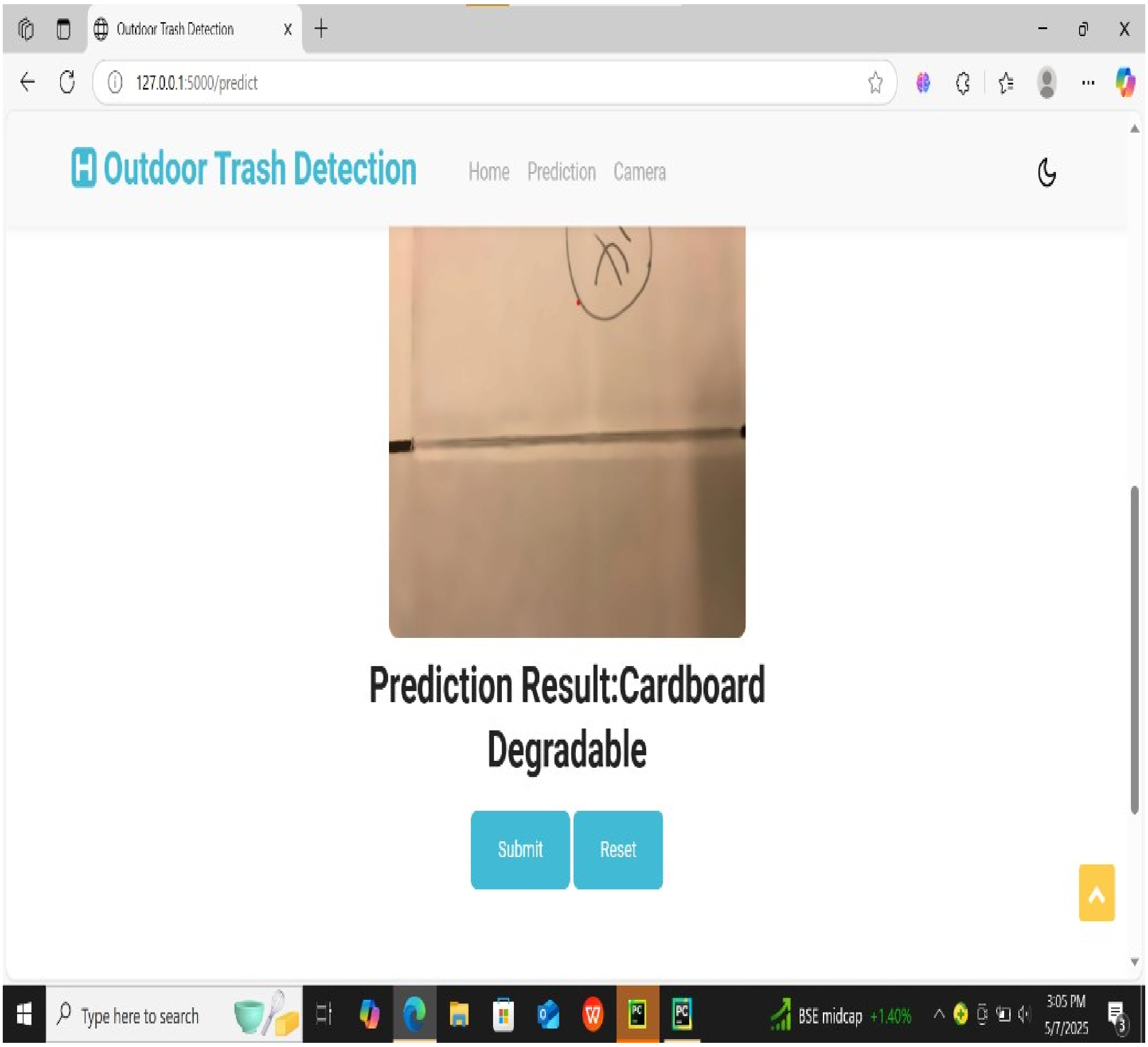

Fig.B.2 Prediction Result 2

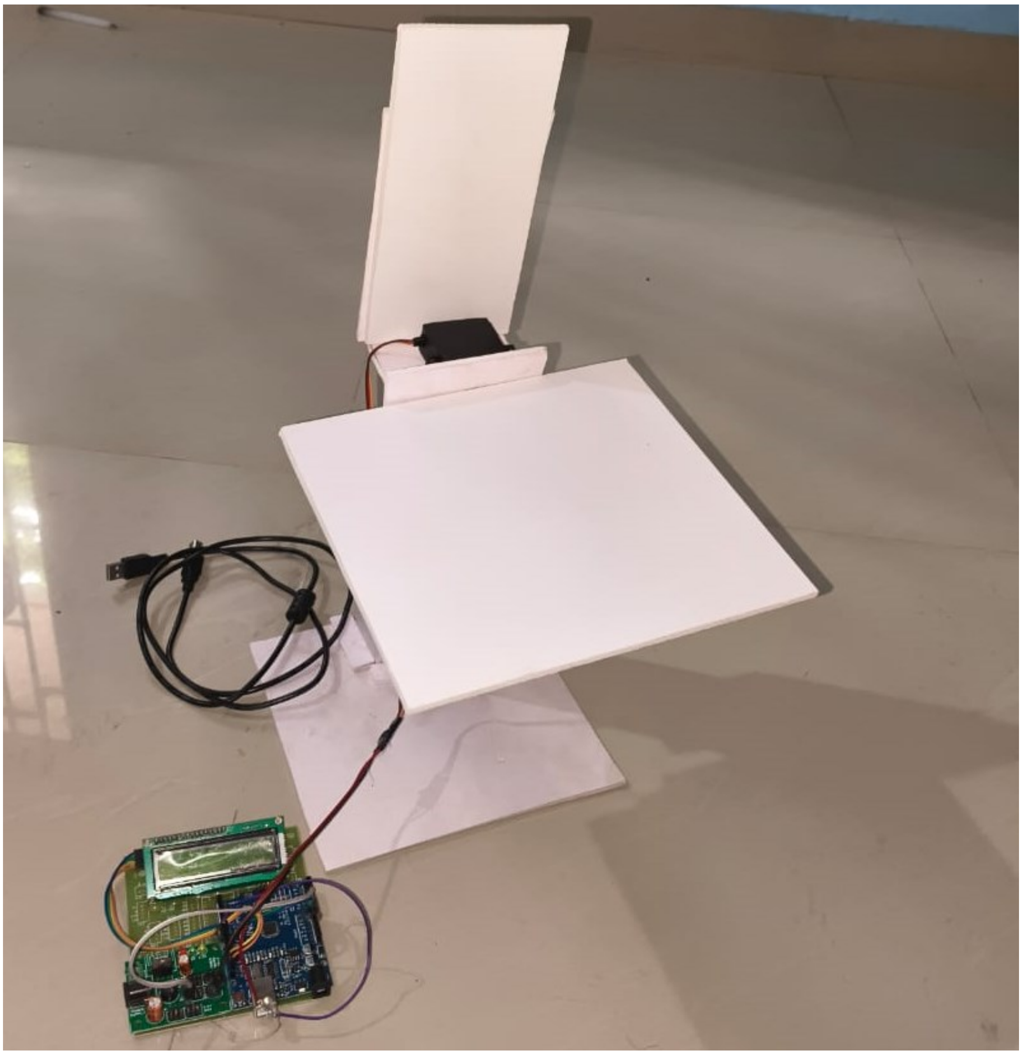

Fig.B.3 Hardware Model

REFERENCES

1. Sami, Khan Nasik, Zian Md Afique Amin, and Raini Hassan. “Waste management using machine learning and deep learning algorithms”.Springer,453-470,2024.

2. Shahab, Sana, Mohd Anjum, and M. Sarosh Umar. “Deep learning applications in solid waste management”, Research Gate,1032-1077,2024.

3. Sarker, Nibir, et al, “Illegal trash thrower detection based on hogswrm for a real-time monitoring system”, IEEE,654-723,2023.

4. Majchrowska, Sylwia, et al, “Deep learning-based waste detection in natural and urban environments”, Research Gate,437-589,2023

5. Nnamoko, Nonso, Joseph Barrowclough, and Jack Procter , “Solid waste image classification using deep convolutional neural network”, Springer,567-692,2023.

6. Toğaçar, Mesut, Burhan Ergen, and Zafer Cömert , “Waste classification using AutoEncoder network with integrated feature selection method in convolutional neural network models”, IEEE,543-678,2022.

7. Lin, Kunsen, et al, “Deep convolutional neural networks for construction and demolition waste classification”, Springer,1045-1132,2022.

8. Spyridakos, Athanasios, et al, “Waste classification of spent refractory materials to achieve sustainable development goals exploiting multiple criteria decision aiding approach”, Research Gate,789-812,2021.

9. Windrim, Lloyd, et al, “Unsupervised ore/waste classification on open-cut mine using close-range hyperspectral data”, IEEE,654-723,2021.

10. Bailey, G. W., M. R. Gilbert, and O. Vilkhivskaya, “Waste classification assessment of nuclear steels for fusion power applications”, Springer,982-1031,2021.