Metropolitan University

Department of Software Engineering

# PROJECT Documentation

## Vegetable Recognition System using ESP32-CAM

Course Title: Embedded Systems & IOT Lab

Course code: SWE-466

### ❖ Submitted to :

Nawshad Ahmed Chowdhury,

Assistant Professor and Head

Dept. of Electrical & Electronics Engineering

Metropolitan University, Bateshwar,
Sylhet - 3104.

### ❖ Submitted by :

Raj Kiron Kishan :- 213-134-016

Tushar Sinha :- 221-134-005

Dipto Deb Nath :- 221-134-013

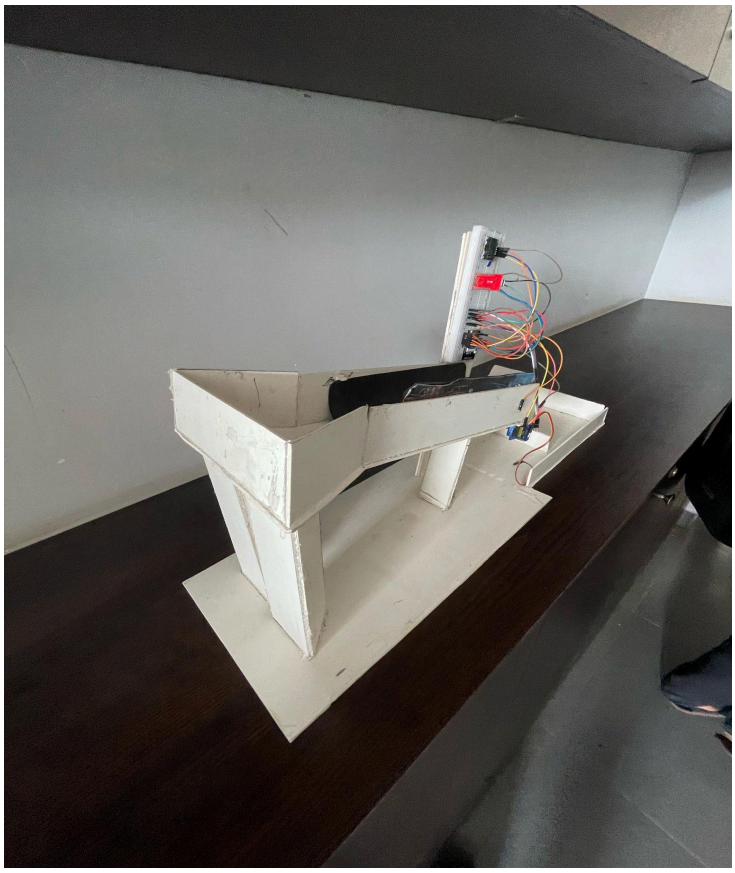
Khondokar Ahmed Mehraz :- 221-134-031

Abhishek Chowdhury :- 213-134-013

## Objective:

The project aims to create an intelligent vegetable recognition system using the ESP32-CAM module. The system of recognition will utilize machine learning models that are executed on the ESP32-CAM in order to classify vegetables such as tomatoes, potatoes, and onions. The result will be displayed on an OLED screen, providing real-time vegetable classification. The project touches on the pillars of edge computing, data sampling, model training, and deployment in Iot applications.

## Project Photographs:



## Components Used:

### Hardware:

1. ESP32-CAM – Captures images and processes classification.
2. OLED Display – Displays the identified vegetable name.
3. FTDI Programmer – Used to upload code to the ESP32-CAM.
4. Resistors & Capacitors – For circuit stability.
5. Breadboard & Jumper Wires – For easy prototyping.
6. 5v Power Supply – Powers the system.
7. Servo Motors - For Sorting.

### Software:

- Arduino IDE – To write and upload code.
- C/C++ – Programming language used.
- Edge Impulse – For training and deploying the machine learning model.
- I2C Library – Enables OLED communication.

## Features:

- **AI-Powered Recognition**: Classifies vegetables in real-time.
- **OLED Display Output**: Shows detected vegetable names.
- **Compact and Portable**: Runs independently without external processing.
- **Energy Efficient**: Low power consumption suitable for embedded applications

### Procedures:

1. Collected image datasets for tomatoes and garlic.
2. Trained a machine learning model using Edge Impulse.
3. Exported and deployed the trained model onto the ESP32-CAM.
4. Assembled the circuit with the OLED display and ESP32-CAM.
5. Uploaded the program to ESP32-CAM and tested the recognition system.
6. Used servo motors to open and close the door at different angles for sorting purposes.

# Problems and Solutions

This entire project was a collective work. Right from learning nothing at the initial phase to completing the project, it was a painstaking but rewarding ride for each and every one of us. Right from purchasing hardware from various stores (offline stores and online sites) to addressing most of the software issues, nothing came smoothly. We laboured collectively, sat collectively, and engaged collectively in solving issues through dialogue, generating ideas, and brainstorming.

The following are the major issues we faced and how we solved them:

---

## 1. Version Issues

**Solved By: Raj Kiron Kishan (213-134-016)**
**Solution:**
I spent an entire day on versioning problems. It was frustrating, but determination made it succeed. Each library needed a different version, and the correct versions needed to be installed before code generation.

---

## 2. Resetting Code on the ESP32-CAM

**Solved By: Raj Kiron Kishan (213-134-016)**
**Solution:**
It took me two days to learn how to reset the code for the ESP32-CAM board. I needed to keep opening and closing the IO0 and GND pins thousands of times until I could reset it properly.

---

## 3. Flash Implementation

**Solved By: Raj Kiron Kishan (213-134-016)**
**Solution:**
It was tough to implement the flash feature. I managed after a lot of trial and error. Because the ESP32-CAM has a poor camera resolution, it was very hard to take training data without flash due to insufficient lighting. The implementation of Flash made this easier.

---

## 4. Object Recognition Issues

**Solved By: Tushar Sinha (221-134-005)**
**Solution:**
On the last submission day, object recognition wasn't working properly because of the camera and environmental conditions. Detection will be impossible without a quality dataset. I had to reconfigure and reset the data six times to get accurate and consistent results.

---

## 5. Display Problems

**Solved By: Abhishek Chowdhury (213-134-013)**
**Solution:**
Our project was highly dependent on object detection and displaying the result on the OLED. The display wasn't functioning initially due to faulty wiring. Once debugging and proper code and wiring connections were provided, the issue was resolved.

---

## 6. Servo Motor Not Working

**Solved By: Khondokar Ahmed Mehraz (221-134-031)**
**Solution:**
The servo motors were not working due to version issues and a poor power supply. Once the wiring issue was fixed and both motors received a good power supply, the problem was solved.

---

## 7. Project Outlook Design

**Solved By: Dipto Deb Nath (221-134-013)**
**Solution:**
Designing the physical layout of the project was among the most troublesome aspects. We had to plan how the ESP32-CAM would be oriented, how objects would roll onto the camera, and how servo-controlled doors would sort vegetables when they were picked up. Upon much planning and adjustment, we were able to develop an effective and functional design.

## Appendix:

**Github link:** [Vegetable Recognition Program]