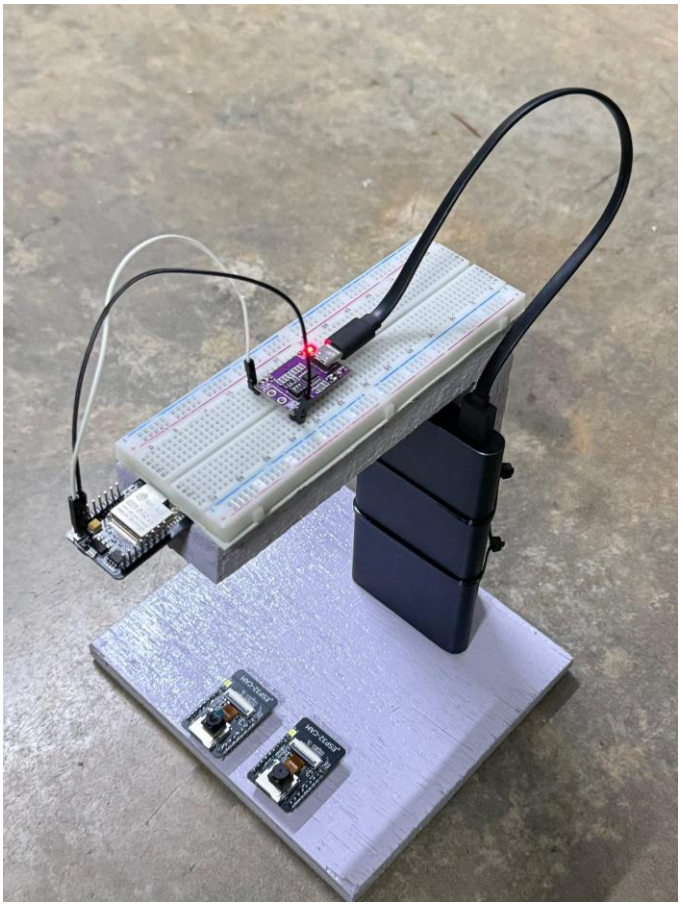
AIROST AI INTERNSHIP REPORT

Team Hokkien Mee is Not Prawn Mee

Members

1. Lee Wei Xuan
2. Looi Yu Xiang



1.0 Introduction

In today's fast-paced world, keeping track of inventory is crucial for efficiency and organization. Our project introduces an **AI-powered automatic inventory management system** designed for drawers. Advanced computer vision and AI algorithms then analyze the images to accurately count the quantity of items inside.

The results are updated in real-time and displayed on a user-friendly website. Users can easily search online to check the real-time availability of specific items without needing to physically open the drawer.

This innovative solution eliminates manual effort, enhances accessibility, and ensures accurate inventory management, making it ideal for various applications, from homes to offices and beyond.

2.0 Timeline

Date	Event
23/10/2024	● Formed project team ● Conducted initial project discussion and brainstorming ● Established project scope and objectives
03/11/2024	● Completed comprehensive research on required hardware components ● Identified necessary technical specifications ● Finalized hardware procurement list
9/11/2023	● Received all hardware components ● Initiated ESP32-CAM programming ● Developed initial codebase for camera functionality
12/11/2023	● Successfully completed first test run ● Achieved camera functionality implementation ● Established WiFi streaming capabilities
17/11/2024	● Implemented facial detection functionality ● Successfully integrated people counting feature ● Validated library integration for detection systems
26/11/2024	● Successfully implemented object detection with custom-trained model ● Completed model training for three ecosystem categories ● Developed and launched initial website interface
01/12/2024	● Successfully deployed complete system ● Integrated website functionality ● Demonstrated working prototype ● Project documentation finalized

3.0 Game mechanics and hardware

3.1 Initial proposed concept

Our project introduces an **AI-powered automatic inventory management system** designed for drawers. The system uses a **camera** to calculate the number of items every time the drawer is opened. Advanced computer vision and AI algorithms then analyze the images to accurately count the quantity of items inside.

The results are updated in real-time and displayed on a user-friendly website. Users can easily search on the website online to check the real-time availability of specific items without needing to physically open the drawer.

During the first proposal, we decided to use Arduino Uno as our server and connect with ESP32-CAM. The Arduino Uno serves as the central controller, managing communication and coordinating actions between components. It can signal when to capture images using the ESP32-CAM and handle data transmission within the ESP32-CAM and send data to the database. The ESP32-CAM module captures images of the drawer's contents. With its built-in Wi-Fi capabilities, it could also be used to transmit images directly for processing if needed.

3.2 Changes made during development

Initially, we selected the **Arduino Uno** as the central controller for its simplicity and cost-effectiveness. However, during development, we found that its processing power was insufficient for handling the image data and AI algorithms required by the system. Its limited speed and memory led to slower performance, which would have hindered the real-time functionality of our solution.

To overcome this, we replaced the Arduino Uno with a **laptop**, which offered significantly higher processing power, enabling smooth execution of AI models and image processing algorithms. This shift not only improved the system's response times but also enhanced overall performance. Using the laptop also provided access to advanced development tools and libraries, making the implementation of our AI-powered inventory management system more efficient.

Additionally, we switched from **Firebase** to **XAMPP with MySQL** for storing and managing inventory data. This change simplified the database setup and allowed for easier configuration and better integration with the system, offering greater control over the data.

For object detection, we initially planned to use **TensorFlow**, but we decided to use **YOLOv11** instead. YOLOv11 offered faster processing and better accuracy for real-time object detection and counting, proving to be a more efficient solution for our project's needs.

3.3 Hardware specifications

The system uses an **ESP32-CAM module** with a 2MP OV2640 camera, built-in Wi-Fi, and 5V power input to capture and transmit images. These images are sent to a **laptop**, which runs the **YOLOv11** model for object detection. The laptop is equipped with a processor, 8GB of RAM, and sufficient storage to handle the image processing.

The **AFC Decoy Trigger** circuit is used to control the power to the ESP32-CAM, providing a stable 5V supply. Communication between the ESP32-CAM and laptop occurs via **Wi-Fi**. The system stores inventory data in an **SQL database** and displays real-time information on a **web interface**.

4.0 Software

ESP32-CAM Code (Developed in Arduino IDE)

The ESP32-CAM is programmed using the Arduino IDE. Its main responsibilities include configuring the camera module for image capture and enabling Wi-Fi communication. The module captures images and sends them to the laptop through HTTP requests, enabling seamless communication.

Laptop Software (Developed in Python)

The laptop serves as the processing hub for the system. It performs the following tasks:

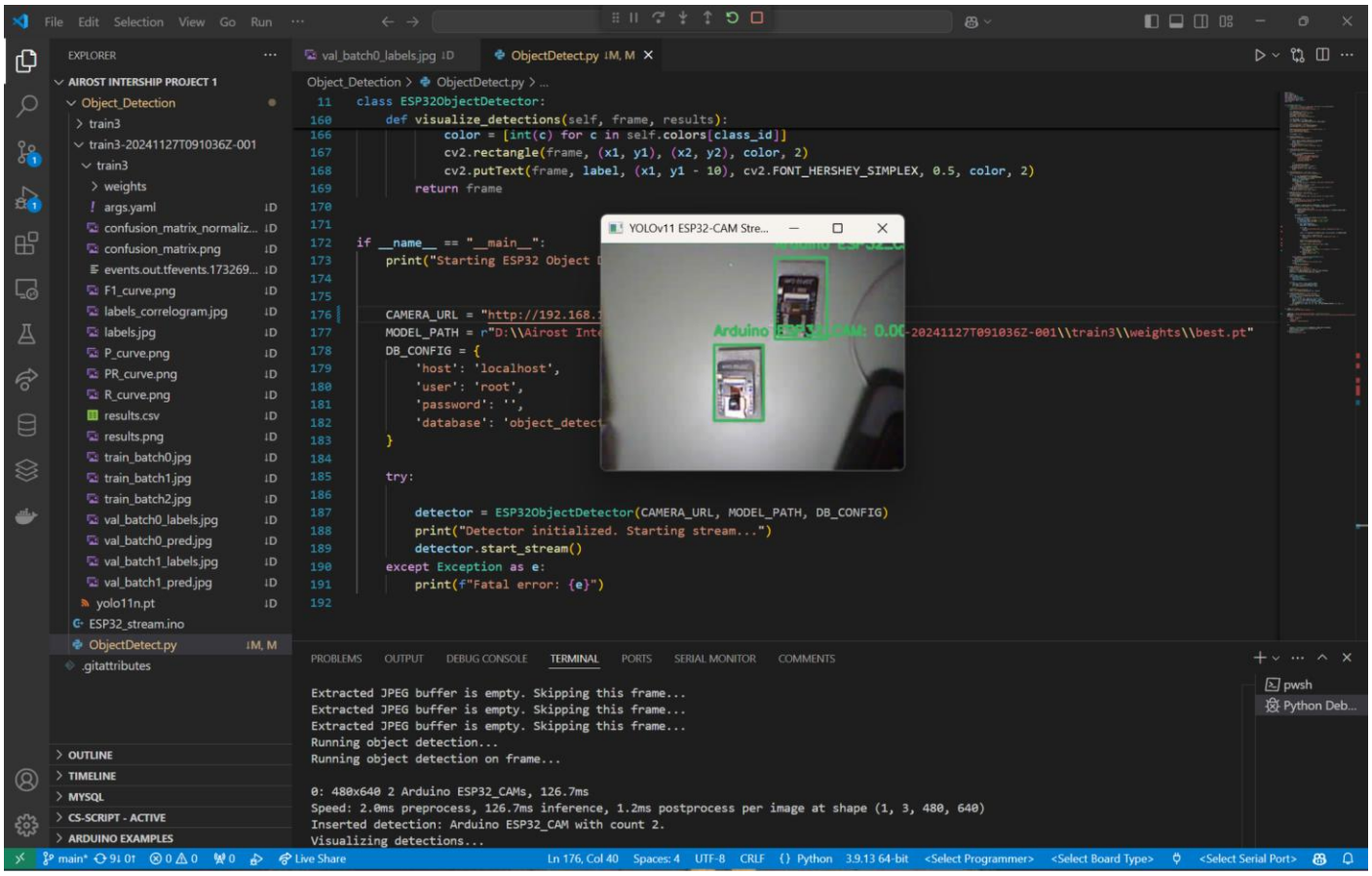
1. Receives images transmitted by the ESP32-CAM over Wi-Fi.
2. Runs the YOLOv11 object detection model using the PyTorch framework to analyze the images and detect objects.
3. Updates the SQL database with the detected objects and their quantities, ensuring that the inventory data is accurate and up-to-date.

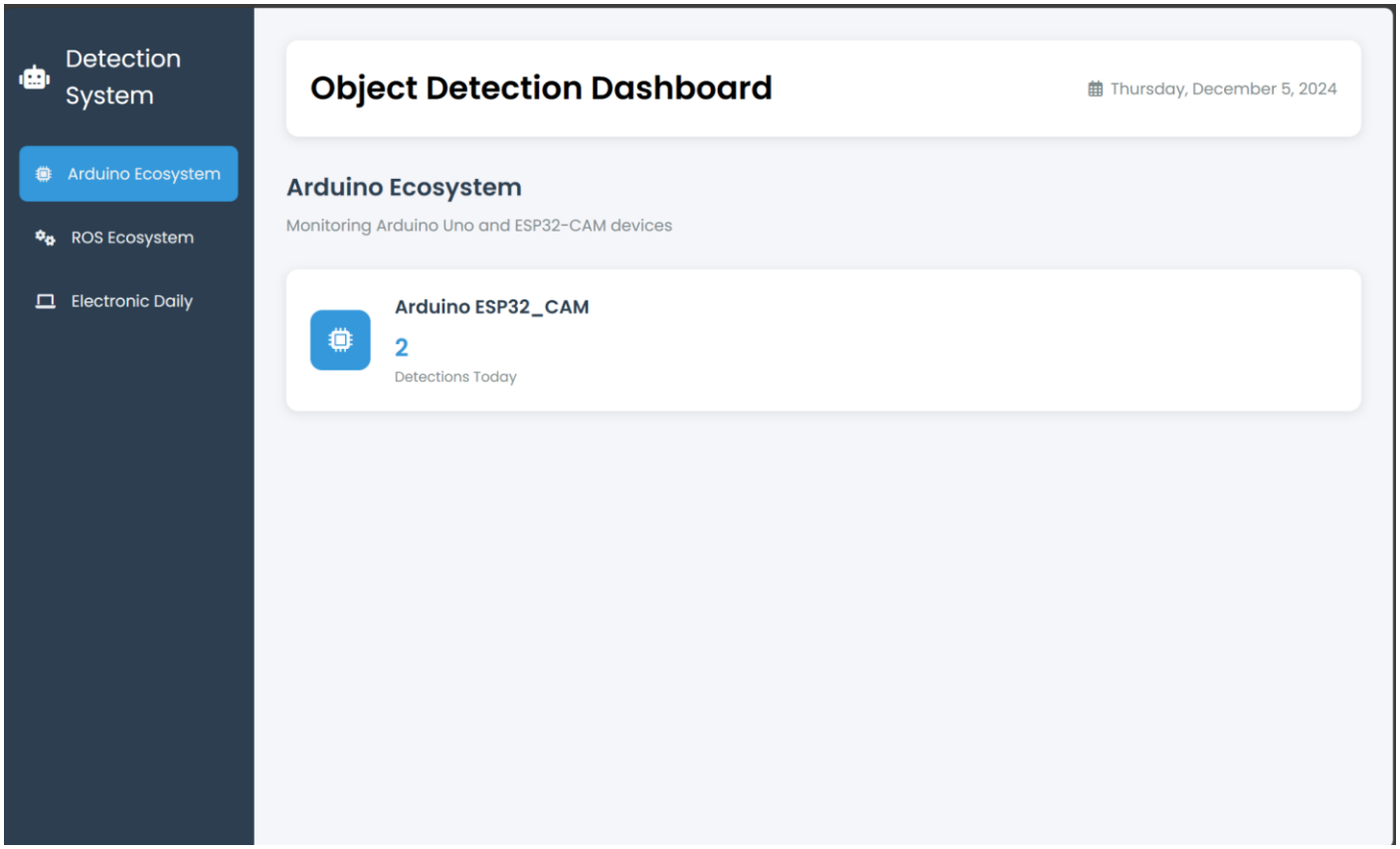
Website Interface

The website is developed using a combination of PHP, HTML, CSS, and JavaScript. It performs the following functions:

1. **Data Retrieval:** PHP scripts query the SQL database to fetch the latest inventory data.
2. **Dynamic Display:** HTML and CSS ensure a clean and user-friendly interface, while JavaScript adds interactivity and responsiveness to the website.
3. **Search and Visualization:** Users can search for specific items, and the website displays their availability in real-time, eliminating the need for manual inventory checks.

5.0 Execution and running





6.0 Conclusion

The project successfully demonstrates a scalable AI-driven inventory management system. By integrating YOLOv11 for object detection and ESP32-CAM modules, the system ensures accurate real-time inventory tracking.

Future improvements include:

- Optimizing YOLOv11 models for faster detection on laptops with limited resources.
- Developing a mobile app interface for enhanced accessibility.
- Adding power optimization for ESP32-CAM modules to ensure longevity.

This solution provides an efficient and accessible platform for inventory management across multiple domains.

REFERENCES

Ultralytics. (2024, November 7). *YOLO11 🚀 NEW*. Ultralytics YOLO Docs. https://docs.ultralytics.com/models/yolo11/

Fahad, E., & Fahad, E. (2024, September 21). *ESP32 CAM with Python OpenCV Yolo V3 for object detection and Identification*. Electronic Clinic. https://www.electroniclinic.com/esp32-cam-with-python-opencv-yolo-v3-for-object-detection-and-identification/

Santos, S., & Santos, S. (2024, June 15). *ESP32-CAM Video Streaming and Face Recognition with Arduino IDE | Random Nerd Tutorials*. Random Nerd Tutorials. https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/

LME Editorial Staff. (2023, March 2). *Getting started with ESP32-CAM*. Last Minute Engineers. https://lastminuteengineers.com/getting-started-with-esp32-cam/

espressif/esp32-camera. (n.d.). GitHub. https://github.com/espressif/esp32-camera