ESP32-CAM + YOLOv8 Real-Time Object Detection

Project Overview

This project combines edge-device video streaming with high-performance object detection. The ESP32-CAM module streams MJPEG video over a Wi-Fi network, which is then processed on a computer using Python and the YOLOv8 deep learning model for real-time object detection. It demonstrates a practical blend of IoT and AI, suitable for smart surveillance, robotics, and automation use cases.

What I Did

- Configured and programmed an **ESP32-CAM (AI Thinker)** to stream MJPEG video using a lightweight HTTP server.
- Connected the ESP32-CAM to a **2.4 GHz Wi-Fi** network and retrieved its streamable IP address using Serial Monitor.
- Wrote and modified Python scripts using **OpenCV** to access the ESP32 video stream.
- Integrated the **YOLOv8 object detection model** (Ultralytics) to analyze each video frame in real time.
- Tested object detection on various items using a local Python environment to simulate a live AI vision pipeline.

Project Breakdown (Cards)

📷 ESP32-CAM Video Stream

The ESP32-CAM acts as a web camera that streams MJPEG video frames through a local IP address over Wi-Fi. It uses a simple HTTP server to make the video feed accessible on any device within the same network.

🧠 YOLOv8 Object Detection

The real-time video feed is consumed by a Python script that runs the YOLOv8 model. It performs object detection on every frame, drawing bounding boxes and labels around detected objects like people, bottles, or phones.

Tech Stack Used

ESP32-CAM (AI Thinker)

A microcontroller with an integrated OV2640 camera, perfect for low-cost video streaming applications.

Wi-Fi Network (2.4 GHz)

Provides the communication bridge between the ESP32 module and the computer for MJPEG streaming.

Python + OpenCV

Python script uses OpenCV to capture and process the MJPEG stream, feeding it frame-by-frame into the YOLO model.

YOLOv8 by Ultralytics

A powerful deep learning model used for real-time object detection with high accuracy and speed.

Outcome

This project successfully demonstrates how to combine low-power edge devices with powerful AI models. It shows that even lightweight hardware like the ESP32-CAM can be used to build real-time, AI-powered vision systems when offloaded to a computer. The result is a flexible and scalable system ideal for learning and prototyping smart vision solutions.