IOT Project:

In this project, I developed a real-time energy conservation system using computer vision and IoT technology. The system optimizes energy usage in environments like classrooms or laboratories by automating the control of electrical appliances based on human activity detected by CCTV cameras. We use a separate server to process video data using the YOLOv5 model for person detection and a Raspberry Pi to control the appliances.

System Architecture

1. **Perception Layer:**
    - **Camera Module:**
        - The camera captures live video footage of the room and streams this data to the server for processing.
2. **Computational Layer:**
    - **Server-Side Processing:**
        - A dedicated server, separate from the Raspberry Pi, receives video streams from the camera.
        - This server runs the YOLOv5 model, which performs real-time human detection on the incoming video data.
        - The YOLOv5 model generates detection results, indicating the presence or absence of people in specific regions of the room.
3. **Network Layer:**
    - **Communication:**
        - The server and the Raspberry Pi communicate over a network using socket programming.

        - The server sends processed data, which includes binary information about human presence in various sections of the room, to the Raspberry Pi.
4. **Application Layer:**
    - **Raspberry Pi Control:**
        - The Raspberry Pi receives detection data from the server.
        - Based on this data, the Raspberry Pi controls electrical appliances like lights and fans through GPIO pins and relays.
        - LEDs are used in the prototype to represent the appliances for simulation purposes.

How It Works

1. **Video Capture and Transmission:**
    - The camera continuously captures video footage and sends it to the server.
2. **Human Detection:**
    - The server processes the video using the YOLOv5 model, detecting human presence in different areas of the room.
    - The server generates a binary array representing human presence in different sections and sends this data to the Raspberry Pi.
3. **Appliance Control:**
    - The Raspberry Pi receives the binary array from the server.
    - It interprets this data to decide which appliances should be turned on or off.
    - The Raspberry Pi controls these appliances via GPIO pins and relays based on the received data.

Implementation

1. **Server-Side Code:**

    - The server captures video frames from the camera and processes them using YOLOv5.
    - It detects human presence and prepares a data packet containing this information.
    - This packet is transmitted to the Raspberry Pi over the network.
2. **Raspberry Pi Code:**
    - The Raspberry Pi listens for incoming data from the server.
    - It interprets the data to manage the state of connected appliances (e.g., turning lights on or off).

Conclusion

The project successfully demonstrates a real-time energy conservation system by leveraging a dedicated server for video processing and a Raspberry Pi for appliance control. This approach ensures efficient energy management and reduces operational costs by automating the control of electrical appliances based on human activity detection.