Grid Mapping Based Smart Electrical Appliances control using IoT

*

Deepak Soni
CSE-AIE
Amrita Vishwavidyapeetham
City, Country
ch.en.u4aie22010@ch.students.amrita.edu

Neelraj R.
CSE-AIE
Amrita Vishwa vidyapeetham
Chennai, India
ch.en.u4aie22035@ch.students.amrita.edu

Niranjan Kumar D
CSE-AIE
Amrita Vishwa vidyapeetham
Chennai, India
ch.en.u4aie22036@ch.students.amrita.edu

Rahul K.
CSE-AIE
Amrita Vishwa vidyapeetham
Chennai, India
ch.en.u4aie22044@ch.students.amrita.edu

Shyam Prasath J
CSE-AIE
Amrita Vishwa vidyapeetham
Chennai, India
ch.en.u4aie22051@ch.students.amrita.edu

***Abstract*—Energy conservation is an important factor used to reduce wastage and meet the environmental impact of power consumption. With the growing demand of power all over the world, there is a need to adopt different measures to optimize the usage of energy. Energy can be effectively saved by only keeping the used electrical appliances active and deactivating others and automating this process without any manual interactions. Computer vision and IoT serves as a new technology by enabling real-time monitoring and control of energy consumption in various environments like classrooms, labs, industries. The proposed system can be easily implemented using existing CCTV cameras without requiring much hardware setups. Through implementation of IoT layers, along with smart algorithms substantial reduction in energy consumption is achieved leading to cost savings.**

***Index Terms*—Internet of Things(IoT), Raspberry Pi, YOLOv5(You look only once version 5), Grid Segmentation, Person detection, Energy management.**

I. Aim

The aim of this project is to implement a computer vision based approach which uses YOLOv5 model for real-time detection of persons in a particular region and control electrical devices in a room to conserve energy when they are not in use. A client-server architecture is proposed where the information from the model(server) is given to the client(Raspberry pi). The client translates the information from the server to control the devices such as lights and fans based on the human activity obtained from the model. This can be directly implemented with CCTV cameras without requiring additional hardware setups, making it easier to implement in various places.

II. Equipment used

The equipments used in the project includes:

- *Camera module:* The camera module used in this project is vital to capture the area where people are present. It provides the input source for the computer vision algorithm, which detects the presence of individuals.
- *Jumper wires:* In this project, the jumper wires are used to make electrical circuit connections within the components. For example, they are used to connect the LED with the raspberry pi GPIO pins
- *Resistors:* The main purpose of the resistors is to limit the current flow passing through the components, such as limiting the current supply to LED to prevent it from getting damaged
- *Bread Board:* The Bread board used in this project helps to make connections between the Raspberry pi board and the other components such as LED and resistors used in this project.
- *Relay:* The relay acts as a switch that is controlled by the Raspberry Pi, which receives data about the presence or absence of individuals in the classroom. When individuals are detected, the Raspberry Pi activates the relay, allowing electrical power to flow to the appliances, thereby turning them on. Conversely, when no individuals are detected, the Raspberry Pi deactivates the relay, cutting off power to the appliances and turning them off.

  .
- *Raspberry pi:* Raspberry pi plays the main role in this project as it obtains the output results from the camera module. Based on the result, Raspberry pi controls the devices accordingly by turning it on or off.

.

- *LED:* LED's are used to mimic the devices in this project. These LED's are controlled by the results given to the Raspberry pi board.

III. Image of Prototype

The sample prototype with the necessary connections for the proposed model is shown in the Figure 1.

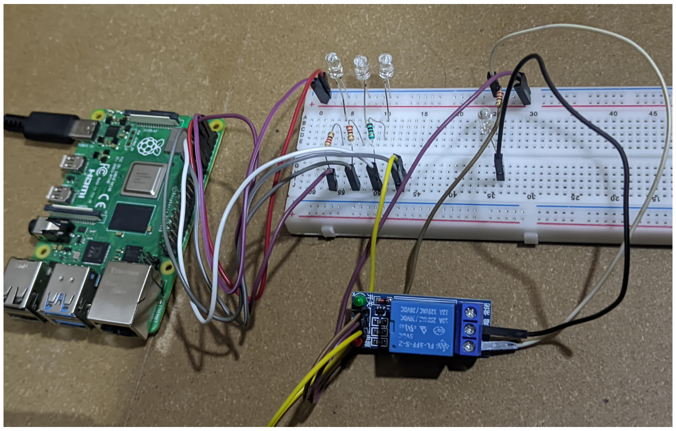

Fig. 1. Image of the prototype

IV. Block Diagram

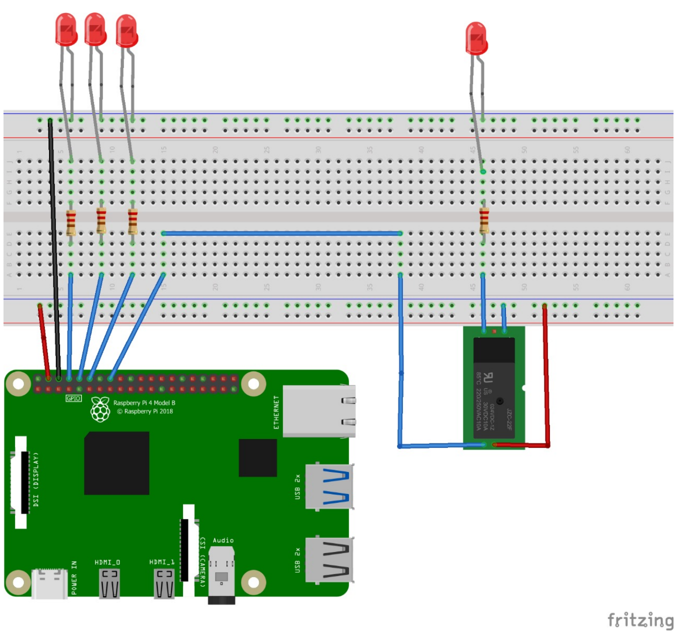

Fig. 2. Circuit Block diagram

V. Working Instances

In this section a detailed explanation of the project details are infused along with the different IoT layers that connects the processes of data collection to model deployment. The workflow of the proposed model and different layers of communication are mentioned in the figure 2.

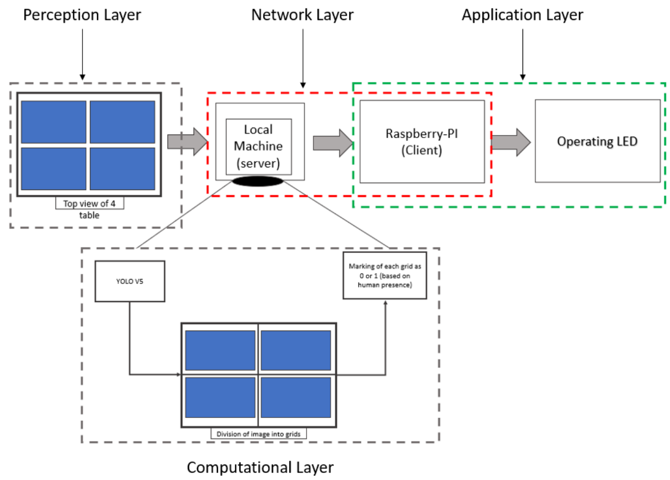

Fig. 3. Architecture Diagram

A. Perception Layer

The perception layer is setup in lab which serves as a interface between the physical setup and the digital system. It consists of process of capturing and transmission.

- Optical sensor:
  The camera is used as the optic sensor which is positioned at the top center with respect to four tables. The camera captures the coverage of entire lab space, allowing for monitoring of dynamic environment.
- Capturing:
  The camera captures the dynamic video footage of the lab including the four tables and other objects pertaining to the frame. This real-time data collection enables perception layer to stay updated about the movement of objects in the lab.
- Transmission:
  Once the data is captured it is then transmitted to the local machine for further analysis. This transmission can occur through wired or wireless communication channels which ensures rapid integration of perception layer with data computational layer.

B. Computational layer

The data computational layer is the crucial part in the local machine which takes raw data as input from the perception layer for further analysis and decision-making.

- Model selection:
  YOLOv5(You Look Only Once Version 5) is the model which is used for effectively detecting the objects within the images. This model is pre-trained on large datasets. YOLOv5 was selected based on its higher accuracy and real-time performance. While previous attempts with HOG-SVM gave lower accuracy levels, YOLOv5 consistently performed well, making it a better choice for data computational layer.

- Model Working Principle:
YOLOv5 represents a cutting edge technology of real-time object detection. The operating principle is efficient in computational in the image in one step rather than multiple steps. Initially the image undergoes computational to improve the picture quality by standardizing its size and pixel values. Then it uses the convolution neural network (CNN) to extract the desired features from the image. Following this process, a detection head identifies bounding objects, predicts their bounding boxes, assigning score for it and calculates class probabilities. YOLOV5's output detects the precise information about the object in a rapid speed.

- Computational Procedure:
The video which is sent is converted to images at a time interval of 10 seconds which confirms any human is present or not over a certain period of time between 2 instances of the image. The raw images captured by camera are divided into four equal portions to facilitate more focused analysis and detection YOLOv5 object detection is applied to each portion of image for the presence of humans alone. This is analysis is done to check whether human is present in each frame or not.

- Prediction Result:
For every portion of an image, binary classification is calculated. If no human is present within the particular region then value of 0 is assigned and if any human is detected in the particular region then value of 1 is assigned. The results of binary classification of each four portions of image are combined to form an array of four binary digits.

- Transmission:
The data collected from the model now serves as the processed data or information that is to be shared via network layer.

C. Network Layer

The network layer enables the TCP/IP communication between local machine (server) and the Raspberry Pi (client) using a client-server mechanism over a local network where the IP address and port number are utilized to identify and establish connection between the devices.

- TCP/IP:
TCP/IP (Transmission Control Protocol/Internet Protocol) is the set of rules governing communication between devices on a network. It breaks data into packets, assigns them addresses, and routes them across the network. TCP ensures packets arrive reliably in order, while IP handles addressing and routing to ensure delivery to the correct destination.
- Establishment of Connection:
Both the local machine (server) and the Raspberry Pi (client) are connected to the same local network. They can directly communicate over the local network without requiring internet.

- Server Initialization:
The local machine initializes as the server by running the server program provided. This program creates a server socket and binds it to the server's IP address (which is the same IP address used for the client) and a specific port number (e.g., 12345). The server program is then ready to accept incoming connections from clients.

- Client Initialization:
The Raspberry Pi initializes itself as the client by running the client program provided. This program creates a client socket and connects it to the server's IP address (which is the same IP address used for the server) and port number. The client program establishes the connection with the server.
- Transmitting Data from Server to Client:
Once the output is processed in the server by detection humans, and the connection is established, the data is about the detected areas is sent to the client with time intervals.

D. Application Layer

The application layer of Raspberry Pi is responsible for controlling the electrical represented by LEDs based on the binary inputs received via network layer.

- Hardware connection:
The Raspberry by is connected to the breadboard through jumper wires which acts as interface for controlling the LEDs representing electrical devices. For visualization purpose this project implemented LEDs while in actual implementation it will correspond to appliances.

- LED Simulation:
Instead of directly controlling the power of fans and lights, LEDs are used for demonstration purposes. With respect to simplified image captured by the camera, four partitions were made. Each partition corresponds to each table. Each table is associated with one light and one fan. So this light and fan are demonstrated as a single unit of source as LED. According to this in total four LEDs were used along with four resistors to control the current flow to the LEDs.

- LED Control and Mapping:
According to the binary inputs from the array the respective LEDs are mapped. For instance, If the binary input is 0, indicating absence of human, the corresponding LED representing the electric device for the table will be turned off. In contrast, if binary input is 1, indicating presence of human, LED will be turned on in order to simulate the activation of electric device.

- Relay:
  In deploying the application layer, direct current (DC) is typically used for simulating the operations of LEDs. However, in order to mimic the real-time conditions of Alternating current (AC) functionality, for one specific LED alone electric relay was used. This relay serves as a important component, allowing LED to perfectly emulate the behavior of devices powered by AC thereby adding an additional layer of realism to the simulation.

VI. Code

A. Person detection using Yolov5s model:

```
import cv2
import torch

# Load the YOLOv5 model
model = torch.hub.load('ultralytics/yolov5', 'yolov5s')

# Set the model to evaluation mode
model.eval()

# Load the image
image_path = 'a3.jpeg'
image = cv2.imread(image_path)

# Perform human detection
results = model(image)

# Process the detection results
for result in results.xyxy[0]:

    if result[-1] == 0:
        x1, y1, x2, y2 = map(int, result[:4])
        cv2.rectangle(image, (x1, y1), (x2, y2),
        (0, 255, 0), 2)

# Display the image with bounding boxes
cv2.imshow('Human Detection', image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

B. Server side:

```
import cv2
import torch
from PIL import Image
import time
import socket

# Initialize YOLOv5 model
model = torch.hub.load('ultralytics/yolov5', 'yolov5s')



def process_video(n,time_interval, video_path, server_socket):

    cap = cv2.VideoCapture(video_path)


    frame_rate = cap.get(cv2.CAP_PROP_FPS)
    total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))

    time_seconds = time_interval * 1.0


    frame_number = int(frame_rate * time_seconds)
    cap.set(cv2.CAP_PROP_POS_FRAMES, frame_number)

    ret, frame = cap.read()

    # Convert frame to PIL image
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    frame = Image.fromarray(frame)

    # Run YOLOv5 inference
    results = model(frame)
    person_results = results.pred[0][results.pred[0]
    [:, 5] == 0]

    # Determine if persons are detected
    if len(person_results) > 0:
        person_detected = 1
    else:
        person_detected = 0



    s=str(n)+str(person_detected)
    server_socket.sendall(s.encode())
    print("Sent:", s)

    cap.release()
    return super

video_streams = [r"E:\fan1-neel.mp4",
                 r"E:\fan3-deepak.mp4"]
time_intervals = [0, 5, 10, 15]


# Server details
SERVER_HOST = '11.12.34.65'
SERVER_PORT = 65432

# Create a server socket
with socket.socket(socket.AF_INET, socket.SOCK_STREAM
                   ) as server_socket:
    server_socket.bind((SERVER_HOST, SERVER_PORT))
    server_socket.listen()
    print("listening for incoming connections")

    # Accept incoming connections
    conn, addr = server_socket.accept()
    print(f"Connected to client at {addr}")

    # Process each time interval for each video stream
    for interval in time_intervals:
        for video_path in video_streams:
            process_video(video_streams.index(video_path),
                          interval,
                          video_path,
                          conn)
            print()
        time.sleep(5)
    conn.close()
```

C. Client Side:

```
import socket
import RPi.GPIO as GPIO

# Configure GPIO mode
GPIO.setmode(GPIO.BCM)

# Set GPIO pin for the bulb

GPIO.setup(14, GPIO.OUT)
GPIO.setup(15, GPIO.OUT)
GPIO.setup(18, GPIO.OUT)
GPIO.setup(23, GPIO.OUT)

# Server details
SERVER_HOST = '11.12.34.65'  # Replace with the actual server's IP addre
SERVER_PORT = 65432

# Connect to the server
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as client_socket:
    client_socket.connect((SERVER_HOST, SERVER_PORT))
    print("Connected to the server.")

    # Receive and process messages from the server indefinitely
    while True:
        try:
            # Receive message from server
            data = client_socket.recv(1024).decode("utf-8")
            if not data:
                break  # Connection closed by server

            # Process YOLO output
            yolo_output = int(data[-1])
            fan = int(data[0])   # Extract YOLO output
```

```
            # If YOLO output is 1, turn on the bulb
            fans = [14,15,18,23]
            if yolo_output==1:
                GPIO.output(fans[fan], True)
                print(f"Bulb turned on in {fans[fan]}")
            else:
                GPIO.output(fans[fan], False)
                print(f"Bulb turned OFF in {fans[fan]}")

        except KeyboardInterrupt:
            print("Client shutting down...")
            break

# Clean up GPIO
GPIO.cleanup()
print("Client closed.")
```

VII. Results and Discussions

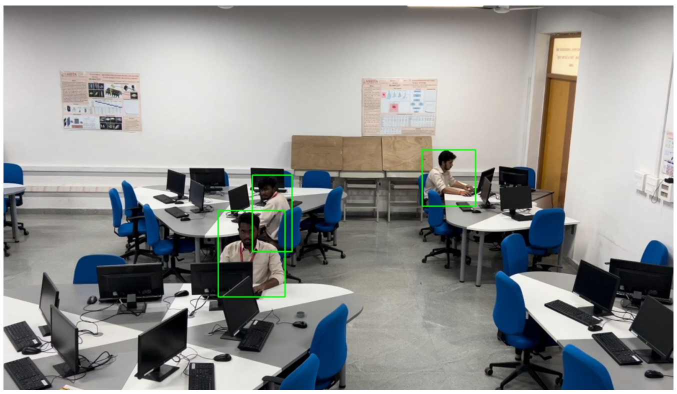

Fig. 4. YOLO person detection

The project was implemented successfully by using LEDs as representation of Fans/lights which were switched on in the presence of human within a designated grid, which shows a significant achievement in development of effective electrical appliance control systems. With LEDs glowing to indicate individuals' presence and the light/fan is turned on, and the process is repeated in every 10 second.

The results shows the effectiveness of the approach, providing clear visual indicators of detected individuals and take needed step. The interval of 10 seconds ensures timely updates which will also minimizing the consumption of resource. However, further optimization in detection algorithm and use of advanced technologies could improve the accuracy of functionality.

Fig. 4 shows the backend working of the person detection. After testing with many images, it was evident that the model performs very well in detecting any humans within the frame with minimal error.

VIII. Conclusion

The proposed system has efficiently integrated the computer vision and IoT technology to improve energy conservation and user convenience in a room by keeping only the necessary electrical appliances active and turning off others automatically. Future works include, making the pereson detection more efficient and to expand device control capabilities to cover a wide range of appliances, adding security layer to the server-client communication and integrating this system on CCTV cameras. Overall, the project shows promising result for intelligent environment management system and sets the groundwork for future advances in this field.