



---

## **Real Time Face Based Energy Monitoring System**

**Deepanshi<sup>1</sup>, Sakshi<sup>2</sup>, Riya<sup>3</sup>, Diksha<sup>4</sup>, Mr.Harkesh<sup>5</sup>**

Department of CSE-AI&DS,

Panipat Institute of Engineering & Technology, Samalkha, India<sup>1,2,3,4,5</sup>

<sup>5</sup>Mail ID: [harkesh.cse.ai-ds@piet.co.in](mailto:harkesh.cse.ai-ds@piet.co.in)

**Abstract**—Unnecessary electricity consumption in classrooms, laboratories, offices, and homes can occur when appliances remain active after occupants leave. This paper presents a software-based real-time monitoring prototype that combines webcam-based face detection with simulated energy consumption to identify possible energy wastage. OpenCV's Haar Cascade classifier detects frontal faces, NumPy generates simulated energy values between 200 W and 600 W, Pygame produces an audible warning, and Matplotlib displays the latest ten readings. An alert is generated when no face is detected and simulated energy exceeds 400 W. The system operates locally for a 40-second monitoring cycle and does not require physical energy sensors or external APIs. The project reports successful operation under suitable lighting, smooth graph updates, and reliable alert playback. Limitations include sensitivity to lighting and face orientation and the use of simulated rather than measured energy data. The prototype provides a low-cost foundation for future sensor-based and IoT-enabled energy management systems [1].

**Keywords:** Face Detection, Real-Time Monitoring, Energy Simulation, Sound Alert, Matplotlib, Smart Room Automation, OpenCV, Python, Presence Detection, Graph Visualization.

### **INTRODUCTION**

The increasing demand for energy and the global focus on sustainability have made efficient electricity usage more crucial. In homes, classrooms, offices, and laboratories, lights, fans, air conditioners, and other appliances may remain active even when rooms are unoccupied. This causes energy wastage, higher utility costs, and unnecessary environmental impact [1].

Advanced smart-room systems commonly employ IoT devices, PIR detectors, smart switches, and cloud-based automation. Such solutions can require hardware cost, installation expertise, stable connectivity, and maintenance, making them less accessible for small-scale or academic demonstrations [1].

The present project proposes a software-only alternative that uses a webcam to detect human presence, simulates energy consumption, and triggers an alert when energy is high while the room is apparently unoccupied. A real-time graph visualizes recent energy readings. The approach is intended as a low-cost educational prototype that demonstrates smart monitoring without physical energy sensors [1].

### **PROBLEM STATEMENT AND OBJECTIVES**

The problem is unnecessary electricity consumption caused by appliances left on in shared environments such as classrooms, laboratories, and small offices. Manual monitoring is unreliable, while sensor-based



automation can be impractical for small demonstrations. The project therefore targets a real-time, software-based room monitoring system that detects human presence using computer vision and alerts users when simulated energy usage is high and the room is unoccupied [1].

- Detect presence through OpenCV and a standard webcam.
- Simulate 200–600 W energy values using NumPy.
- Check whether energy exceeds the 400 W threshold when no face is detected.
- Generate an audio alert through Pygame.
- Visualize the latest ten readings using Matplotlib.
- Run the complete monitoring process for a 40-second cycle.

### LITERATURE REVIEW

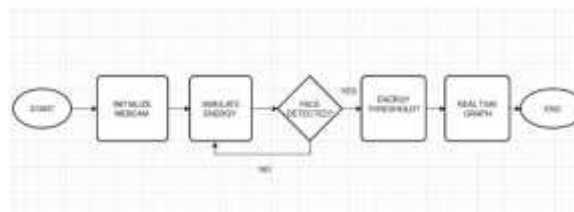
Viola and Jones introduced a boosted cascade approach for rapid object detection using simple features [2]. This work motivates the use of Haar Cascade detection in lightweight real-time applications. Redondi et al. reviewed object-detection algorithms on embedded platforms and emphasized computational efficiency [3].

Ma et al. discussed existing-building retrofits and practical challenges in improving building infrastructure [4]. Ali et al. presented Python-based smart energy monitoring, supporting the use of Python as an accessible platform for energy-related processing [5]. Hunter introduced Matplotlib for scientific visualization [6]. Gulati and Joshi examined sensor-based safety systems with sound alerts [7]. Kumar and Tiwari addressed vision-based automation in smart buildings [8], while Mohan and Sharma discussed simulated energy monitoring in IoT applications [9].

### PROPOSED METHODOLOGY

#### A. System Architecture

The proposed system integrates five functional modules: webcam capture, face detection, energy simulation, alert evaluation, and graph visualization. The webcam supplies frames to the detection module. OpenCV estimates occupancy, NumPy provides a virtual power reading, and a rule-based decision module combines occupancy and energy state. Matplotlib maintains a short history of readings [1].



*Fig. 1. Flow chart of the proposed monitoring process [1].*

#### B. Face Detection

Each webcam frame is converted to grayscale and processed using the Haar Cascade frontal-face classifier. Detection of one or more faces indicates occupancy; absence of a detected face indicates apparent vacancy. The project notes that performance is affected by shadows, low illumination, side angles, and occlusion [1], [2].



### C. Energy Simulation

Because the prototype has no physical energy meter, NumPy randomly generates integer values from 200 W to 600 W. The simulated values represent a virtual appliance load and allow the complete alert and visualization pipeline to be tested. They should not be interpreted as measured electricity consumption [1].

### D. Alert Decision

Let  $P$  be the occupancy indicator and  $E$  the simulated energy. The threshold is  $T = 400$  W. The alert condition is:

$$A = 1 \text{ if } P = 0 \text{ and } E > 400 \text{ W; } A = 0 \text{ otherwise.}$$

### E. Real-Time Visualization

The system stores up to ten recent energy values and updates a Matplotlib line graph after each new reading. This gives users immediate visual feedback on energy variation and possible high-consumption periods [1], [6].

**TABLE I. SOFTWARE TOOLS USED IN THE PROPOSED SYSTEM [1].**

| Module         | Tool/Library       | Purpose                               |
|----------------|--------------------|---------------------------------------|
| Face detection | OpenCV             | Haar Cascade real-time face detection |
| Video input    | cv2.VideoCapture() | Capture webcam feed                   |
| Simulation     | NumPy              | Generate random energy values         |
| Audio alert    | Pygame             | Play .wav warning sound               |
| Graph plotting | Matplotlib         | Plot live energy data                 |
| Language       | Python 3.8+        | System logic and integration          |

### SYSTEM WORKFLOW AND ALGORITHM

The application initializes the webcam, Haar Cascade model, audio file, graph, and 40-second timer. During each iteration it captures a frame, converts it to grayscale, detects faces, generates a simulated energy value, evaluates the alert condition, updates the graph, and waits before repeating [1].

1. Initialize webcam and libraries.
2. Load the Haar Cascade model and alert sound.
3. Start the 40-second monitoring timer.
4. Capture a frame and convert it to grayscale.
5. Detect faces and determine occupancy.
6. Generate a 200–600 W simulated energy value.
7. If no face is detected and  $E > 400$  W, play the warning sound.
8. Append the reading and update the latest-ten graph.
9. Repeat until the time limit expires.
10. Release the webcam and terminate.

### EXPERIMENTAL SETUP AND IMPLEMENTATION

The prototype was evaluated with different users and lighting conditions during a 40-second monitoring cycle. The test exercised webcam capture, face detection, energy generation, decision logic, audio



playback, and graph updates together. The evaluation was intended as functional validation rather than a statistically rigorous benchmark [1].

#### A. Software Environment

The implementation uses Python with OpenCV, NumPy, Pygame, and Matplotlib. The system operates locally and does not require cloud services or external APIs. A standard webcam is used as the input device [1].

#### B. Process Flow

The process begins with system initialization, followed by continuous acquisition and processing. The face detector produces the presence state, while the simulated energy generator supplies the load value. The decision logic then determines whether an alert is required.

#### C. Experimental Conditions

The project reports that face detection was effective in suitable lighting and frontal poses. Performance decreased in low light, with side profiles, and when faces were partially occluded. Simulated energy values remained within the 200–600 W range. The audio warning was triggered when the defined vacancy and threshold conditions occurred, and the graph was updated with recent values [1].

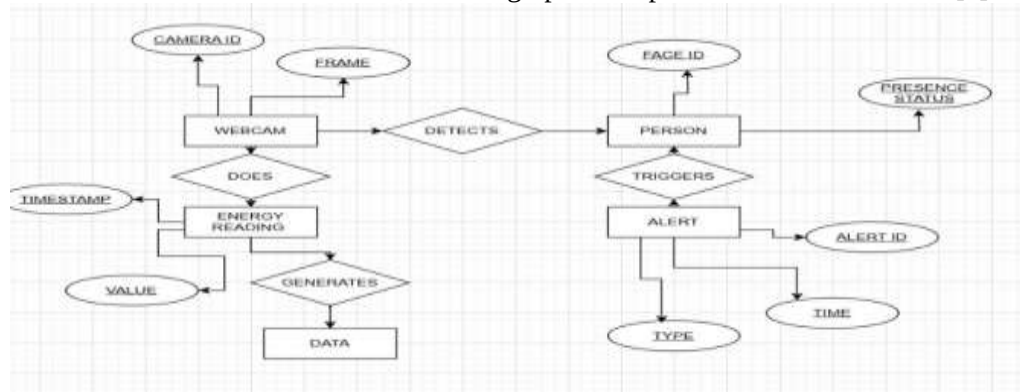


Fig. 2. Data-flow representation of the monitoring system [1].

## RESULTS

#### A. Face Detection Results

The Haar Cascade detector successfully identified visible frontal faces under favorable conditions. Missed detections occurred in challenging lighting or pose conditions. Because the system interprets no detected face as vacancy, missed detections may result in false energy-wastage alerts [1].

#### B. Energy Simulation and Alert Results

The energy generator produced values between 200 W and 600 W approximately every 3–4 seconds. The 400 W threshold was used to identify high simulated consumption. The project reports reliable triggering of the audio warning, with an alert duration of approximately three seconds [1].

#### C. Visualization Results

The Matplotlib graph maintained the ten latest readings and updated dynamically. The project reports smooth updates without noticeable lag during the 40-second cycle, allowing users to observe energy spikes and lower-use periods [1].

**TABLE II. SUMMARY OF EXPERIMENTAL OBSERVATIONS [1].**

| Aspect              | Observation                    | Remarks                       |
|---------------------|--------------------------------|-------------------------------|
| Face detection      | High accuracy in good lighting | Sensitive to shadows and pose |
| Energy simulation   | 200–600 W random values        | Demo values, not measured     |
| Alert system        | Reliably triggered             | Immediate local attention     |
| Graph visualization | Smooth real-time updates       | Shows energy patterns         |
| System stability    | 40 s without crash             | Suitable for prototype        |

### RESULTS AND DISCUSSION

The integrated prototype demonstrates that computer vision, data simulation, alerting, and visualization can be combined into a single low-cost monitoring application. Its main functional contribution is the linkage between an occupancy estimate and a virtual energy stream, followed by an immediate user-facing warning when the room appears vacant while energy is high [1].

The results should be interpreted within the scope of the prototype. The supplied report does not provide numerical precision, recall, F1-score, or a calibrated energy-saving percentage. Therefore, the reported results establish functional operation rather than quantified electricity reduction.

A missed face may be interpreted as vacancy even when a person is present. Conversely, detecting a face does not prove that every appliance is being used appropriately. Occupancy is therefore a useful proxy for energy justification, not a complete measure of room activity.

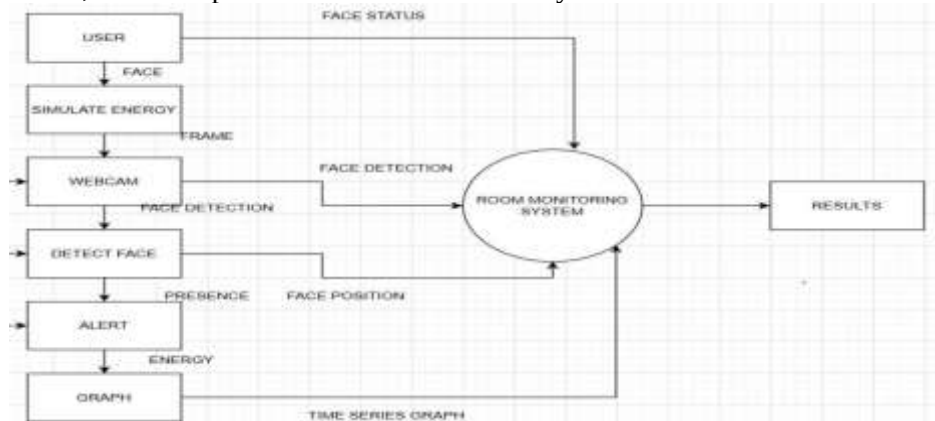


Fig. 3. Detailed data-flow diagram of inputs, processing, alerting, and visualization [1].

### LIMITATIONS

| Limitation               | Description                                           |
|--------------------------|-------------------------------------------------------|
| Lighting dependency      | Detection accuracy decreases in low-light conditions. |
| Front-facing requirement | Side-angle or occluded faces may be missed.           |
| No real sensor input     | Energy values are simulated rather than measured.     |
| Local audio only         | No remote email or mobile notification.               |
| Fixed runtime            | Current cycle is limited to 40 seconds.               |
| No appliance control     | System does not switch real devices.                  |



---

These limitations mean that the current system is best considered a proof of concept. For production deployment, measured energy data, robust occupancy estimation, longer-duration validation, privacy-aware processing, and safe appliance-control interfaces would be necessary.

### **COMPARATIVE DISCUSSION WITH RELATED WORK**

The project adopts the lightweight detection philosophy represented by classical object detection [2] and the computationally accessible approach emphasized in embedded detection surveys [3]. Unlike hardware-heavy smart-room systems, it intentionally avoids physical energy sensors and IoT infrastructure. This reduces setup cost but also limits the system to simulated energy values. The use of Python libraries for simulation, visualization, and alerting makes the prototype particularly suitable for academic demonstration [5], [6].

The literature also indicates that smart-building systems can benefit from vision-based automation [8] and IoT-based energy monitoring [9]. The present project can therefore be interpreted as an initial software layer that can later be connected to those hardware and communication components.

**XI. CONCLUSION**  
This paper presented a Real Time Face Based Energy Monitoring System designed to identify possible electricity wastage in apparently unoccupied rooms. The system integrates webcam input, OpenCV Haar Cascade face detection, NumPy-based energy simulation, Pygame audio feedback, and Matplotlib visualization. An alert is generated when no face is detected and simulated energy exceeds 400 W [1].

According to the project report, the prototype achieved its fundamental objectives during the 40-second monitoring cycle. Face detection was effective under suitable lighting, simulated energy values remained within the intended range, audio alerts were produced under the defined conditions, and graph visualization operated smoothly [1].

The work demonstrates that a basic smart-room energy-awareness concept can be implemented without dedicated IoT hardware. At the same time, simulation and simple face detection cannot provide the same reliability as calibrated sensors and robust occupancy models. The system should therefore be viewed as a low-cost proof of concept rather than a production building-management platform.

### **FUTURE SCOPE**

- Integration with real energy sensors and IoT-enabled smart plugs to obtain accurate consumption data and enable actionable control.
- Improved occupancy detection using modern deep-learning detectors such as YOLO, SSD, or MobileNet-SSD and multi-person tracking.
- Cloud connectivity for historical logging and remote notifications through authorized services.
- A GUI or web dashboard showing live video, occupancy, energy readings, alerts, thresholds, and historical trends.
- Long-term data collection and machine learning for anomaly detection, forecasting, and energy-saving recommendations.
- Multi-room and multi-camera deployment for classrooms, offices, laboratories, and smart campuses.
- Alternative visual/mobile alerts in addition to local audio.
- Integration with smart switches or relays for automatic appliance control, with safety checks and manual override [1].



---

## REFERENCES

- [1] Riya, Sakshi, Diksha, and Deepanshi, “Real Time Face Based Energy Monitoring System,” Project Report (Project-II), Dept. of CSE (AI & DS), Panipat Institute of Engineering and Technology, Samalkha, Panipat, India, May 2025.
- [2] P. Viola and M. Jones, “Rapid Object Detection Using a Boosted Cascade of Simple Features,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2001, doi: 10.1109/CVPR.2001.990517.
- [3] A. Redondi, L. Baroffio, M. Cesana, and M. Tagliasacchi, “Object Detection Algorithms Running on Embedded Platforms: A Survey,” Future Internet, vol. 8, no. 3, p. 35, 2016, doi: 10.3390/fi8030035.
- [4] Z. Ma, P. Cooper, D. Daly, and L. Ledo, “Existing Building Retrofits: Methodology and State-of-the-Art,” Energy and Buildings, vol. 55, pp. 889–902, 2012, doi: 10.1016/j.enbuild.2011.08.021.
- [5] I. Ali, F. Siddiqui, and S. Ahmed, “Smart Energy Monitoring System Using Python,” International Journal of Advanced Research in Computer Science, 2018.
- [6] J. D. Hunter, “Matplotlib: A 2D Graphics Environment,” Computing in Science & Engineering, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.
- [7] R. Gulati and D. Joshi, “Smart Safety System Using Sensor Network and Sound Alerts,” International Journal of Computer Science and Software Engineering, 2021.
- [8] R. Kumar and P. Tiwari, “Vision-Based Intelligent System for Automation in Smart Buildings,” International Journal of Engineering and Advanced Technology, vol. 8, no. 6S, 2019.
- [9] G. Mohan and P. Sharma, “Simulated Energy Monitoring Systems in IoT Applications,” International Journal of Engineering Trends and Technology, vol. 69, no. 7, pp. 65–70, 2021.