International Journal for Modern Trends in Science and Technology
Volume 12, Issue 05, pages 331-336
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol12issue05.html
DOI: https://doi.org/10.5281/zenodo.20404086

IOT-based Gas Leak Detection and Fire Hazard Recognition System Using Image Processing for Household Safety Enhancement

Mudigonda Swapna, Dr. G Srinivas

Department of Electronics and Communication Engineering, Sri Indu College of Engineering & Technology, Ibrahimpatnam, Hyderabad

To Cite this Article
Mudigonda Swapna & Dr. G Srinivas (2026). IOT-based Gas Leak Detection and Fire Hazard Recognition System Using Image Processing for Household Safety Enhancement. International Journal for Modern Trends in Science and Technology, 12(05), 331-336. https://doi.org/10.5281/zenodo.20404086

Article Info
Received: 25 April 2026; Revised: 19 May 2026; Accepted: 23 May 2026.

KEYWORDS

Internet of Things (IoT), ESP32, Machine Learning, Gas Leakage, Visual Verification, ESP32-CAM, Sensor Fusion.

ABSTRACT

Fire incidents and gas leakage events represent serious safety concerns in residential, commercial, and industrial settings, frequently resulting in significant damage to property and threats to human life. Conventional detection systems generally operate using predefined sensor threshold values, which can increase the possibility of false alerts due to environmental variations such as temporary temperature changes or harmless gaseous emissions. This study presents an advanced Intelligent Safety System that improves traditional IoT-based monitoring through the integration of Machine Learning (ML) techniques and visual verification mechanisms. The proposed system employs an ESP32 microcontroller to analyze information collected from a multi-sensor setup consisting of gas, flame, and temperature sensors through a sensor fusion approach for accurate hazard prediction. In addition, the system integrates an ESP32-CAM module capable of capturing real-time images or providing live video streams when abnormal conditions are detected, allowing users to obtain immediate visual evidence of fire or gas leakage events. The developed solution overcomes the limitations of conventional systems, including blind notifications and reliance on network availability, while delivering a low-cost, scalable, and reliable monitoring framework with enhanced verification capabilities.

1. INTRODUCTION

In the modern era, technological advancements have led to the development of intelligent systems that enhance human safety. The Internet of Things (IoT) has emerged as a revolutionary technology that enables real-time monitoring and data-driven decision-making. One of the most significant applications of IoT is in the domain of safety, where the timely detection of hazards can prevent major accidents.

Gas leakages (LPG, Methane) and fire outbreaks are time-sensitive hazards. A delay in detection or response can lead to catastrophic outcomes. Traditional systems, such as standalone smoke detectors or manual safety inspections, are often limited by their inability to provide instant remote alerts or detailed environmental contexts.

Although recent IoT solutions have introduced remote monitoring via platforms such as Blynk or ThingSpeak, they typically suffer from two major limitations:

False Alarms: Most systems use simple conditional logic (e.g., if Gas > 400ppm, trigger alarm). This fails to account for benign events, such as cooking fumes or cigarette smoke.

Lack of Visual Evidence: Users receiving a remote alert have no way to confirm the severity of the situation without being physically present, often leading to panic or ignored warnings.

This study presents a novel system that upgrades the traditional architecture. By employing Edge AI (running directly on the ESP32) to analyze patterns in the sensor data and integrating an ESP32-CAM for visual proof, the system significantly reduces false positives and ensures a reliable and verifiable response mechanism.

2. LITERATURE REVIEW

The evolution of fire and gas detection has progressed from manual observation to sophisticated electronic monitoring systems.

A. Historical Background

Early detection methods were entirely manual and relied on human observation or simple mechanical tools. The 20th century saw the introduction of electrochemical sensors, but they lacked automation. The advent of microcontrollers and GSM modules in the early 2000s allowed for SMS-based alerts; however, these systems were limited in scalability and lacked data visualization.

B. Existing Works

Sharma et al. (2017) developed a gas leakage detector using Arduino and MQ-2 sensors. Although it successfully activated local buzzers and exhaust fans, it lacked remote monitoring capabilities.

Gupta et al. (2019) designed an IoT-based system using NodeMCU and ThingSpeak. This improved user awareness through cloud data logging but suffered from instability during periods of network congestion.

Singh et al. (2021) proposed an AI-integrated system for predictive detection of faults. However, this system was complex and did not offer visual verification of hazards.

C. Limitations and Research Gaps

The current literature highlights a gap in reliability. Existing systems often face false alarms owing to environmental fluctuations and sensor drift. Furthermore, they are network dependent, meaning that if Wi-Fi fails, the intelligence of the system often fails.

This study addresses these gaps by implementing local AI processing (Edge Computing) and adding a camera module to provide the missing visual link.

3. SYSTEM ARCHITECTURE

The proposed system architecture is divided into three main layers:

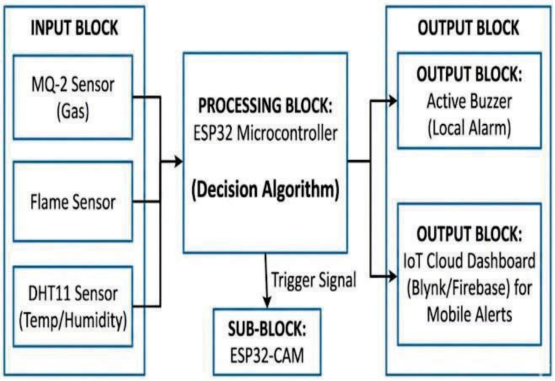
Perception Layer (Hardware)

Intelligence Layer (AI/ML)

Application Layer (Cloud & Visuals)

A. The Block Diagram

The block diagram helps to understand how the system works.



B. Hardware Design

The hardware subsystem is responsible for sensing and generating local alerts.

1. Microcontroller (ESP32)

Selected over the older ESP8266/NodeMCU due to its dual-core processor, which is required to handle the computational load of the ML algorithm and the camera stream simultaneously.

Fig 3.2: Microcontroller(ESP32)

2. Sensors

MQ-2 Gas Sensor: Operates on the Metal Oxide Semiconductor (MOS) principle to detect combustible gases such as LPG, methane, and smoke.

Fig 3.3: MQ-2 Gas Sensor

Flame Sensor: Detects infrared radiation emitted by fire sources.

Fig 3.4: Flame Sensor

DHT11 Sensor: Measures ambient temperature and humidity to provide environmental context for the AI model.

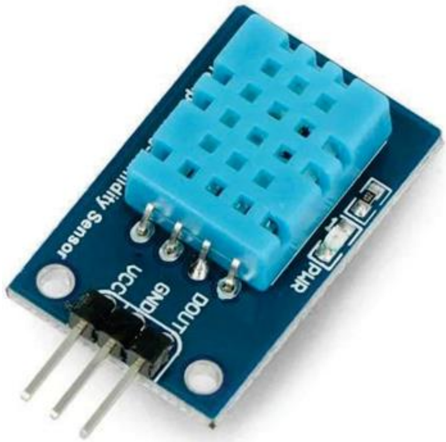


Fig 3.5: DHT11 Sensor

3. Visual Module (ESP32-CAM): A dedicated camera module (OV2640) integrated to capture high-resolution images when a “Critical” state is predicted by the AI.

Fig 3.6: ESP32-CAM

4. Alert Module: A 5V Active Buzzer for local audible alarms.

Fig 3.7: Alert Module

C. Software & Cloud Specifications

Programming Environment: The Thonny IDE (Python) was used for firmware development.

AI Training: Python (Scikit-Learn) was used to train the classification model on a PC, which was then ported to the ESP32.

IoT Platform: The system utilizes a robust cloud backend (e.g., Firebase or upgraded Blynk) to handle both telemetry data and image storage, offering real-time dashboards on mobile devices.

4. METHODOLOGY: AI & VISUAL VERIFICATION

This section details the primary innovation of this study: moving from threshold-based detection to intelligent verification.

A. Data Collection and Sensor Fusion

In a standard system, a gas sensor triggers an alarm if the value exceeds limit X. In the proposed system, Sensor Fusion is utilized.The system continuously collects a vector of data: V = [Gasppm, TempC, Humidity%, FlameIR]

B. Machine Learning Model: To reduce false alarms, a Supervised Learning model (Decision Tree Classifier) is implemented.

The model was trained to classify the state of the room into three categories:

1. Normal: Low gas, normal temperature.
2. Warning (False-Positive Check): High gas (e.g., alcohol fumes or cooking), but normal temperature and no flame IR.

Action: Notification only.

3. Critical (Fire/Leakage): High gas + rising temperature or flame detected.

Action: Full Alarm + Camera Trigger.

This predictive approach addresses the limitation of sensor sensitivity varying depending on environmental conditions.

C. Visual Verification Logic: The ESP32-CAM operates in deep-sleep mode to conserve power.

Logic Flow

The primary ESP32 runs an ML inference loop.

IF output == Critical:

Signal is sent to ESP32-CAM.

The camera wakes up and captures a burst of three images.

The images are uploaded to the Cloud Storage bucket.

A link to the image is sent to the user's smartphone.

This allows the user to immediately differentiate between a real fire and a sensor error.

5. IMPLEMENTATION

A. Circuit Diagram and Connections: The implementation follows a modular design.

Sensors: The MQ-2 analog output connects to the ADC pin (VP) of the ESP32.

The digital output of the Flame sensor was connected to a GPIO interrupt pin to ensure an immediate reaction to the fire.

Camera Interface: The ESP32-CAM communicates with the main controller via serial communication (UART) or a simple logic trigger pin.

Power Supply: A regulated 5V DC source was used to ensure the stability of the Wi-Fi module, which consumes significant current during transmission.

Algorithm Workflow

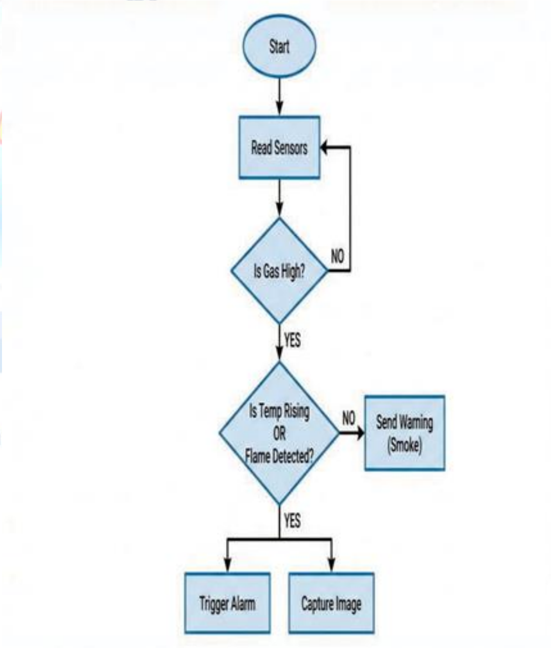


Fig 5.1: System Flowchart

Step 1: Initialize the system, connect Wi-Fi, and warm up the sensors.

Step 2: Read Analog/Digital values (G, T, F).

Step 3 (AI Step): Pass the values to the predict_hazard(G,T,F) function.

Step 4: If Safe: Update Cloud Dashboard.

If Warning: Send Push Notification "Check Air Quality."

If Danger:

Activate Buzzer.
Trigger Camera Capture.
Send Priority Alert with Image Link.
Step 5

Wait for 2 seconds (Sampling Rate) and repeat.

C. IoT Dashboard Setup The application layer was configured using the Blynk IoT Platform.

The dashboard was designed to show:

Gauges: Live visualization of Gas PPM and Temperature.
Image Widget: A dynamic window that refreshes when a new image is captured by ESP32-CAM.

6. RESULTS AND DISCUSSION

A. Performance Comparison

The proposed intelligent system was compared against a basic threshold-based system.

Table 6.1 Basic vs Proposed System

Feature	Basic System (Source)	Proposed Intelligent System
Logic	Threshold (If Gas > Limit)	ML Classification (Sensor Fusion)
False Alarm Rate	High (triggers on cooking fumes)	Low (filters non-hazardous fumes)
Visual Proof	None (Blind Alert)	**Yes (ESP32-CAM Images)**
Latency	Low	Low (< 2 seconds)
User Confidence	Low (uncertainty of risk)	High (visual confirmation)

B. Scenario Testing

Test Case 1: Gas Leakage: LPG was released near the sensors. The MQ-2 value increased rapidly. The system correctly identified this as leakage. The camera captured the cylinder area, allowing the user to remotely observe the pipe condition.

Test Case 2: Interference: A lit cigarette was placed near the sensor. The Gas PPM increased, but the temperature remained stable, and the flame sensor was inactive.

The Basic System triggered a fire alarm. The Proposed System classified this as a "Smoke-Warning" but did not trigger the fire siren or camera, successfully filtering the false positive.

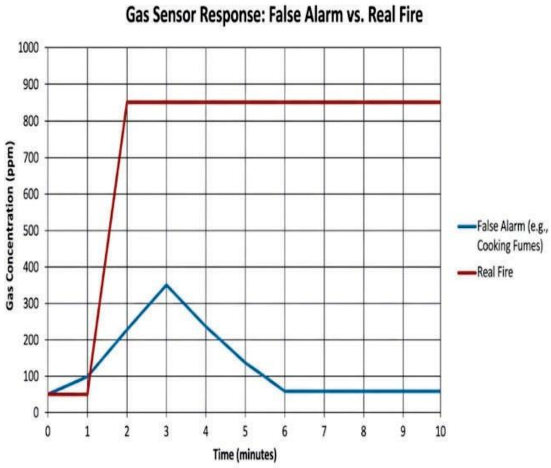
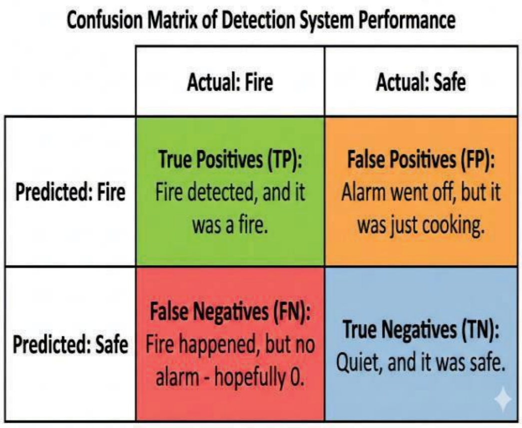


Fig 6.1: Graph showing false alarm vs real fire

Confusion Matrix
True Positives: Fire detected, and it was a fire.
False Positives: Alarm went off, but it was just cooking.
True Negatives: Quiet, and it was safe.
False Negatives: Fire occurred, but no alarm.



7. CONCLUSION

This study presents the design and implementation of a smart IoT-based safety system. By identifying the limitations of existing studies, specifically false alarms and lack of remote visibility, this study successfully integrated Edge AI and ESP32-CAM technology.

The result is a robust system that detects hazards with greater accuracy through sensor fusion and provides actionable visual intelligence.

The system is cost-effective and scalable, making it suitable for widespread adoption in smart homes and industrial safety networks.

The future scope includes implementing object detection directly on the camera to autonomously identify fire shapes, further reducing reliance on cloud processing.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Gavaskar, K., Malathi, D., Ravivarma, G., & Arulmurugan, A. (2021). Development of LPG leakage detection alert and auto-exhaust system using IoT. 7th International Conference on Electrical Energy Systems (ICEES), 558–563.

[2] Adekunle, A. A., et al. (2024). Design and development of a fire alert system using GSM technology. International Journal of Engineering Research & Technology (IJERT), 13(2), 45–50.

[3] Suresh, M., & Kumar, P. (2023). GSM-based gas leakage detection system with SMS alert. Journal of Electronics and Communication Systems, 8(1), 12–18.

[4] Hassan, R., & Ali, S. (2023). Arduino-based LPG gas leakage detection and prevention system using GSM. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 12(4), 1102–1108.

[5] Gupta, S., & Sharma, L. (2023). Implementation of microcontroller-based fire and gas detection system with SMS notification. International Journal of Computer Applications, 185(4), 12–16.

[6] Dhiman, L., Chauhan, P., Singh, M., & Kestwal, P. A. (2024). Enhancing safety measures with an IoT-based fire and gas detection system. International Journal of Enhanced Research in Science, Technology & Engineering, 13(4), 162–166.

[7] Rahman, M., & Islam, T. (2022). Low-cost automatic fire alarm system using GSM and Arduino. Journal of Embedded Systems and Applications, 9(3), 22–29.

[8] Choi, H., & Park, S. (2021). Radio Frequency (RF) based remote fire alarm system for large warehouses. Journal of Logistics and Supply Chain Management, 14(2), 55–62.

[9] Oluwasegun, O., & Adewale, A. (2022). Design of a PIC microcontroller-based gas leakage detector. African Journal of Engineering Research, 10(1), 1–7.

[10] Babu, K. S., & Reddy, V. P. (2023). IoT-based smart gas leakage monitor using NodeMCU. International Journal of Innovative Technology and Exploring Engineering, 12(5), 45–49.

[11] Rao, P. V., & Rekha, B. S. (2023). Wireless gas sensor network for home safety using IoT. Proceedings of the 2023 IEEE International Conference on IoT in Social, Mobile, Analytics and Cloud, 34–39.

[12] Hussain, M., & Ali, T. (2023). Fire detection using ensemble learning and IoT sensors. Journal of Ambient Intelligence and Humanized Computing, 14, 1122–1130.

[13] Wu, L., Chen, L., & Hao, X. (2021). Multi-sensor data fusion algorithm for early indoor fire warning based on BP neural network. Information, 12(2), 59.

[14] Al-Shammari, M., & Al-Qurishi, M. (2022). Design of a smart fire detection system based on IoT and cloud computing. Journal of King Saud University - Computer and Information Sciences, 34(6), 3056–3064.

[15] Roy, S., & Biswas, A. (2022). Standalone LPG gas leakage detector using MQ-6 sensor. International Journal of Sensors and Sensor Networks, 10(2), 25–30.

[16] Obayuwana, A., Olah, D., & Akinbohun, S. (2024). Enhancing fire safety through IoT-enabled flame detection systems: A cost-effective and scalable approach. Proceedings of the Nigerian Academy of Science, 17(1): 54–67.

[17] Tan, W., & Lee, J. (2023). Development of a smoke detection system using MQ-2 sensor for residential use. Journal of Safety Engineering and Management, 6(1), 14–20.

[18] Gopal, T., & Krishna, M. (2022). Temperature-based fire alarm system using LM35 and Arduino. International Journal of Electronics and Communication Engineering, 15(3), 201–206.

[19] Yadav, D., & Kumar, S. (2022). Automatic exhaust fan controller for gas leakage detection. International Journal of Applied Engineering Research, 17(2), 115–119.

[20] Khan, Z., & Ahmed, F. (2023). Simple fire alarm circuit using thermistor and buzzer. Journal of Electrical Engineering Projects, 5(1), 5–9.

[21] Jumaa, N. K., Abdulkhaleq, Y., Nadhim, M., & Abbas, T. (2022). IoT-based gas leakage detection and alarming system using Blynk platform. Iraqi Journal for Electrical and Electronic Engineering, 18(1), 64–70.

[22] Mahmoudi, S., Gloesener, M., Benkedadra, M., & Lerat, J.-S. (2025). Edge AI system for real-time and explainable forest fire detection using compressed deep-learning models. Proceedings of the 20th International Joint Conference on Computer Vision, Imaging, and Computer Graphics Theory and Applications, 847–854.

[23] Khan, M. A., & Kim, J. (2025). Dempster Shafer-empowered machine learning-based scheme for reducing fire risks in IoT-enabled industrial environments. IEEE Access, Early Access.

[24] Ayranci, A. A., & Erkmen, B. (2024). IoT-based fire detection: A comparative study of machine learning techniques. NOHU Journal of Engineering Sciences, 13(4), 1298–1307.

[25] Ullah, Z., & Al-Turjman, F. (2024). Gas leakage detection using Tiny Machine Learning. Electronics, 13(23), 4768.

[26] Kaur, A., & Singh, S. (2025). Optimizing fire safety: Reducing false alarms using advanced machine learning techniques. International Journal of Fire Science, 12(1), 45–52.

[27] Al-Bahadili, H., & Rahma, A. (2023). Smart home safety system using machine learning and IoT. International Journal of Electrical and Computer Engineering, 13(2), 1890–1898.

[28] Sawant, S., et al. (2024). Integrated fire detection system using ML and IoT. International Journal for Research in Applied Science and Engineering Technology (IJRASET), 12(5), 2091–2100.

[29] Talaat, F. M., & ZainEldin, H. (2023). An improved fire detection approach based on YOLO-v8 for smart cities. IEEE Access, 11, 12345–12356.

[30] Al-Hajri, M., & Al-Mansoori, S. (2025). Enhancing fire alarm systems using edge machine learning for smoke classification. Applied Sciences, 15(1), 24.

[31] Espressif Systems, "ESP32 Series Datasheet," 2022.

[32] Hanwei Electronics, "MQ-2 Gas Sensor Technical Data," 2016.