Real-Time Fire Image and Location Dispatch Network utilising Edge Cloud Integration, Twilio, and Telegram

I. Badari Karthik [1] , Dr. V. Shanmukha Rao [2], J.Balu [3]
Department of (CSE-AI)[1], Department of(CSE-IOT)[2], Department of MCA[3]
PSCMR College of Engineering and Technology, Vijayawada
Email: badarikarthikinguva3561@gmail.com, shanmukharao.v@gmail.com, balu.j@pscmr.ac.in,

ABSTRACT

Fire and gas leakage incidents can have massive detrimental effects on human life, property and the environment if not detected. The current conventional methods depends only on local alarm or manual monitoring through windows or doors, which is cumbersome in terms of access from remote locations and may delay emergency services to provide assistance. The proposed system solves this problem by enabling remote monitoring and controlling of fire and gas safety using multiple communication channels simultaneously. Using a combination of the Internet of Things (IoT), edge computing, cloud computing and messaging applications, the proposed real-time fire image and location dispatch network provides a multi-channel communication platform for rapid detection of hazards. This paper proposes an edge–cloud computation -based Fire and Gas Safety System (FGPSS) with a method to dispatch the fire hazard location, image and text to the remote device through available multiple communication channels. An ESP32 Dev Kit V1 is used in the edge computing. The flame and MQ-2 gas sensors continuously monitor and detect fire and gas condition. A NEO-6M GPS module is used to capture the real-time GPS location. The image captured from the ESP32-S3 camera module is sent to the cloud dispatcher written in Google Apps Script via HTTP POST. Duplicate webhook calls are prevented through a unique code. The image is uploaded to cloud storage and the address is sent to the dispatcher. The dispatcher sends the emergency notification using Twilio SMS, Voice Calls, Telegram Bot API and Email to the responsible authority and user. The system also provides a web- based dashboard with the help of Firebase Realtime Database. The administrator can view the incidents in real time and remotely controller the safety devices. The proposed system is reliable, intelligent and scalable. It can be deployed in all smart buildings, industries, educational institutes, residential and smart city environments.

KEYWORDS

Internet of Things (IoT), Fire Detection, Gas Leak Detection, Edge Computing, Cloud Computing, Firebase Realtime Database,

1.INTRODUCTION

Buildings and laboratories, commercial facilities, and public spaces are all susceptible to fire or gas leakage accidents, which may cause severe injury and cause major loss of property and even casualties. Many of the conventional safety systems use local alarms and manual monitoring. Usually, human intervention is needed before any further action can be undertaken. Consequently, emergency incidents may not be reported to the relevant authorities immediately that will result in delayed response and more severe consequences. Recent development in technologies of Internet of Things (IoT), cloud computing, and embedded system provides the opportunities for deploying new smart and efficient solutions on safety and security. Technologies of IoT and cloud enable the deployment of safety solution with the ability of sensors to continuously monitor environmental, data process at the edge, and cloud for remote communication. The intelligent safety solution can automatically detect dangerous situations, notify emergency, and reduce the requirement of safety vigilance. Using a Pyrolyzed food waste, this study aims to evaluate the performance and effectiveness of a CO2-based gas sensor in detecting the odor and greenhouse gases (GHG) emissions from food waste fermentation. Food waste slurry was prepared from the obtainment of food waste at the Seoul Food Waste Company and the Siwon Park Food Waste Company. The slurry was aerated and the gases were added with CO2 and CO to the headspace of the fermenter, which was placed in a sealer bag. The headspace gases were collected and then determined using a GC-POC. The artificial odorants used in this study were carboxylic acids and alcohols. The performance of a CO2-based sensor was determined by the sensitivity and response time of the sensor to artificial odorants and the GHG. By these experiments, it was concluded that CO2-based gas sensors had a high sensitivity and response time to the artificial odorants and GHG. To address these challenges, this project proposes an Automated Cloud-Dispatched Fire and Gas Safety Hub, an IoT-based emergency response system designed for fast and reliable hazard detection. The system uses an ESP32 as the edge computing device to collect sensor data, determine the location of an incident using GPS The cloud dispatcher sends notifications using several channels, such as SMS, voice call, Telegram messages, and bulk email alerts, so the emergency data is promptly delivered to relevant parties. With an ESP32-S3 camera, pictures of the attack scene are taken, while a Firebase-based web dashboard allows the system to be monitored and managed from a remote location and controlling safety devices if necessary. Integrating embedded hardware, cloud services, and communication API into a unified service, this paper proposed a safety solution. The solution provides an efficient, scalable and intelligent service for emergencies detection, fast alert and remote monitoring and control in residential, commercial, industrial and institutional environments

Key Objectives

The core aim of this work is to come up with a dependable real-time fire and gas leak detection system based on IoT and edge-cloud technologies. The proposed system constantly monitors environmental conditions, captures emergency photographs, and detects the exact GPS location of the emergency. The system emits immediate alerts via Telegram, Twilio SMS, voice calls and email for swift emergency response. The system also provides remote monitoring and control of safety devices via a cloud-based web dashboard to improve public safety, reduce response time, and minimize damage incurred during a gas or fire emergency.

• To develop an IoT-based system that detects fire and gas leakage in real-time using ESP32, flame and MQ- 2 gas sensors, respectively.

• To acquire and upload emergency images automatically using ESP32-S3 camera module for evidential purposes. To enable real-time GPS-based location tracking and identification of the emergency site.

• To send real-time multi-channel alerts (Telegram, Twilio SMS, voice calls and email) at the time of the emergency.

• To combine the use of multi-processor (edge and cloud) computing for faster data processing, data storage and real-time synchronization.

• To provide a web dashboard for remote control and monitoring of safety gadgets (exhaust fan and warning light).

• To reduce emergency response time by providing real-time alerts and notifications, live monitoring and automated actions for mitigation of crisis.

1.2 Problem statement

Early warning and quick response is necessary for Fire & gas leakage incidents. The following problems may exist in available safety devices: - Only local alarms; - Remote viewing, controlling of the device; - High time delay in informing emergency contacts and police in case of a fire accident; - Support of visual verification in case of a fire accident; - Device's communication methods: sound alarms only; - Remote control of the controlling device in emergency cases; - A large number of services on small embedded devices when handling numerous communication service demands. Therefore, it is necessary to develop a safe, real-time, location aware and remotely controlled safe and alarm system through Internet. The cloud-dispatched automated fire and gas safety Hub will integrate Internet of Things (IOT) sensors with edge computing and internet processing to make sure quickest possible reporting of events to responsible parties and the quickest possible response.

2.Literature Survey

The literature survey reviews existing research on IoT-based fire detection and emergency response systems using sensors, cloud computing, and image processing. Previous studies provide real-time fire detection and alert notifications but often lack integrated image evidence, GPS tracking, and remote monitoring capabilities. Recent advancements have improved detection accuracy through edge-cloud computing and intelligent communication platforms. Based on these findings, the proposed system combines real-time monitoring, image capture, GPS location, multi-channel alerts, and remote device control into a unified fire safety solution.

S.No	Authors	Contribution	Limitations
1.	Xue Liang Yang et al. (2024)	Developed an edge-cloud fire detection system using RGB and infrared cameras for real-time fire monitoring and cloud storage.	Does not provide multi-channel emergency alerts or remote hardware control.
2.	Agusriyanda Saputra et al. (2024)	Proposed an ESP32-based IoT fire detection system with flame and MQ-2 gas sensors, Telegram alerts, and an automatic sprinkler.	Lacks image capture, GPS location tracking, and cloud-based monitoring.
3.	Deni Kurnia et al. (2023)	Designed a wireless fire detection system using ESP8266 with flame, gas, and temperature sensors for Telegram notifications.	Limited to sensor-based detection without visual evidence or remote control.

4.	Rizqi Sukma Kharisma et al. (2025)	Introduced an AI-based fire detection system using ESP32-CAM and YOLOv8 for image-based fire recognition.	Does not integrate GPS tracking or comprehensive emergency communication services.
5.	P. Vimala Imogen et al. (2024)	Developed a real-time fire surveillance system using YOLOv8, Twilio, React.js, and MongoDB for intelligent fire monitoring.	Requires high computational resources and lacks integrated gas leakage detection and remote device control.

3.RESEARCH GAPS

Based on the literature survey, the current fire detection systems are either based on sensors or image-based, but do not have a comprehensive emergency response system. There are very few solutions that combine real-time image capture, GPS location tracking, multi-channel alert systems and remote device control into one. Moreover, the scalability and responsiveness of the system are limited due to the limited use of edge-cloud integration. Due to these areas of research gaps, it was proposed that the Real-Time Fire Image and Location Dispatch Network, which offers an integrated, intelligent and real time fire safety solution, be developed.

4.PROPOSED METHODOLOGY

The proposed system is an edge–cloud based real-time fire and gas safety monitoring system. The ESP32 Dev Kit V1 is constantly monitoring the data from the flame sensor and MQ-2 gas sensor, and will warn about dangerous situations. In case of fire or gas leakage, the ESP32-S3 camera will take photos of the fire and the NEO-6M GPS module will record the precise location of the detected fire or gas leakage. The data collected is sent to the ESP32 which then sends it to the cloud server through the Google Apps Script (GAS) HTTP API. Information is stored on the cloud and automatically pushed out through Twilio SMS, Voice Calls, Telegram and Email to the Firebase Realtime Database. The monitoring and remote control of safety devices such as warning lights and exhaust fans are real-time, thanks to the web dashboard. After emergency is resolved, it is continued to monitor for future emergencies.

4.1 Hazard Detection

The MQ-2 gas sensor and flame sensor are used to continuously monitor the environment, which is the first step in the hazard detection process. The MQ-2 sensor is used for combustible gases, smoke and gas leakages detection, the flame sensor is used for fire detection. This ESP-32 Dev kit continuously reads and processes the sensor value in real time. If the measurement of any sensor surpasses the preset value, the ESP32 is aware of it and immediately triggers the fire/gas leakage response.

4.2 Image Capture

If fire or gas leakage is detected, the ESP32-S3 Camera module captures a picture of the incident automatically. This provides visual evidence of the incident. The picture is uploaded to Cloudinary a cloud-based image hosting service where the image is securely stored. Once the upload is complete, Cloudinary creates a secure image URL and sends it to the ESP32 Dev-Kit. The same image URL is passed in the emergency

4.3 GPS Location Acquisition

The NEO-6M GPS module always provide in real time the coordinates (latitude/longitude) of the monitored position. The ESP32 Dev Kit will translate the coordinates and transmit a link to Google maps for the quick identification of the immediate emergency position. The GPS information is also transmitted in the alert message, so that users and emergency services are able to accurately identify the location of the emergency event and immediately respond.

4.4 Cloud Processing

Once sensor data, GPS location, and image URL are retrieved, the ESP32 Dev Kit storing them in a JSON file for ease of data transfer. Google Apps Script then receives a JSON data via an HTTP POST request. Google Apps Script also validates the information it receives, processes the emergency information, processes it for storage in the cloud database and activates the emergency notifications for real-time emergency notification.

4.5 Alert Notification

Once the emergency information is processed, the cloud dispatcher will immediately start sending the notification via several communication channels such as Telegram Bot, Twilio SMS, Twilio Voice Call, and Email Service. These alerts are communicated to registered users and emergency responders in a timely manner in the event of an emergency. The content of each notification will contain details on the type of hazard, the location on Google Maps GPS co-ordinates and a photo of the hazard as a visual verification of fire incidents, so that the recipients can decide on action and take appropriate steps.

4.6 Real-Time Database Synchronization

The processed emergency data is stored in the central cloud repository (Firebase Realtime Database). The web dashboard is automatically updated in real-time with the newest alert data, sensor readings, and hardware status in Firebase. This allows administrators to keep track of the current situation, get live updates on the systems and receive the latest updates without any major delay.
Using AI Text?

4.7 Remote Monitoring and Control

The Web Dashboard enables administrators with real-time emergencies alerts and system status. It supplies all information about the hazard detected, real time video stream, GPS location and sensor readings. With the alerts received, administrators control safety equipment remotely from the dashboard, which can have a prompt impact in reducing the impacts of the emergency and overall safety.

4.8 Hazard Mitigation

Admin control commands are sent to Firebase Realtime Database and synced with the ESP32 Dev Kit when sent by an administrator via the Web Dashboard. These commands are passed to the ESP32 which then turns on the relay module for the safety devices, including the exhaust fan and warning light. These devices will keep running until removal of the hazardous condition. The system automatically resets the alerts and restarts the continuous environmental monitoring once the fire or gas leakage has been removed, guaranteeing the protection of the environment in case of future emergencies.

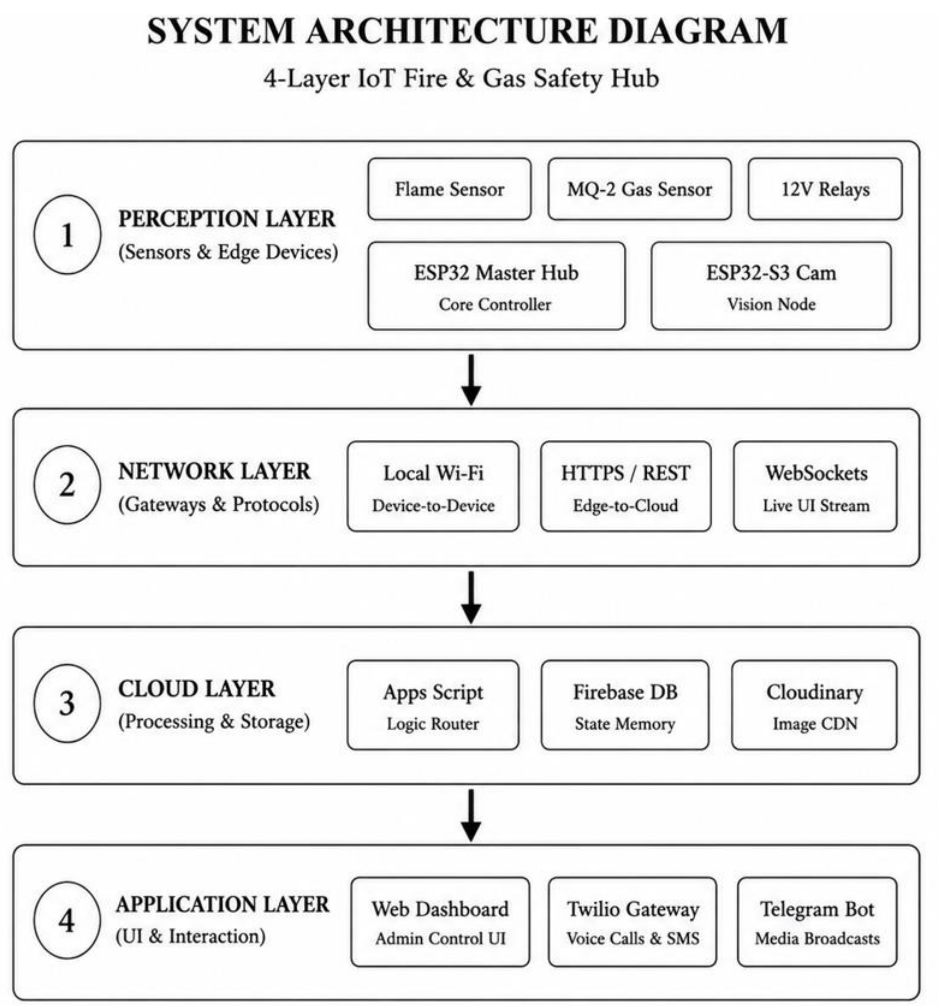
Overall Workflow

- Monitor flame and gas sensor readings continuously.
- Detect fire or gas leakage when the threshold is exceeded.
- Capture the fire image and retrieve the GPS location.

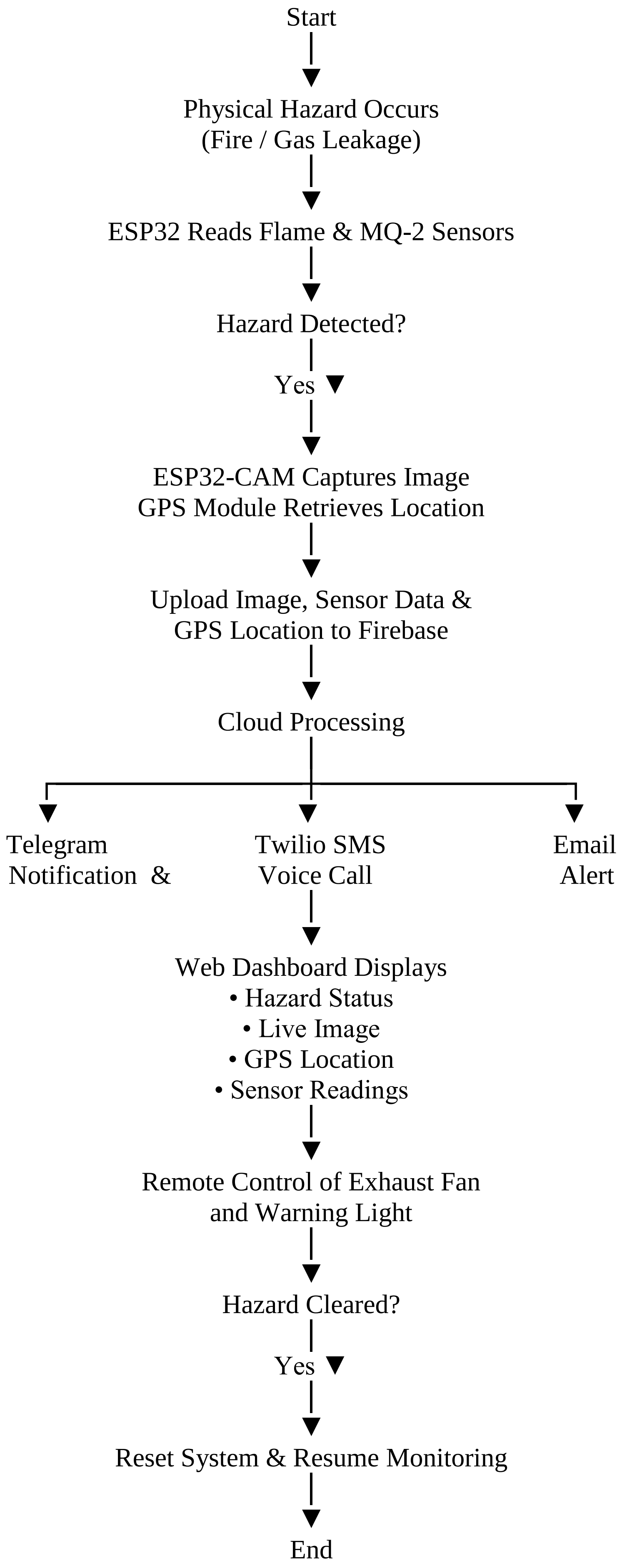
- Upload sensor data, image, and location to the cloud.
- Send instant alerts through Telegram, Twilio SMS, voice calls, and email.
- Display live hazard status and enable remote device control through the web dashboard.
- Reset the system after the hazard is cleared and resume continuous monitoring.

3. SYSTEM ARCHITECTURE

The Real-Time Fire Image and Location Dispatch Network Utilizing Edge-Cloud Integration, Twilio, and Telegram is an edge-cloud integrated solution for real-time fire and gas leakage detection, monitoring and emergency dispatch. The MQ-2 gas sensor, the flame sensor, and the GPS module (NEO-6M), are all connected at the edge layer where data is continuously read by the ESP32 Dev Kit V1. The ESP32 recognizes the emergency if the sensor value measured is above the set threshold, and immediately takes a picture of the affected area using the ESP32-S3 Camera. The captured picture is uploaded to Cloudinary which returns a secure URL to the picture. The ESP32 encapsulates Sensor values, GPS data, hazard type and URL of the picture in an object in json format. This JSON object is then forwarded to Google Apps Script via an HTTP POST message to be processed further. The Google Apps Script gets the object, processes it and saves it to a Firebase Realtime Database. The cloud then sends a number of emergency notification services - Email Service, Twilio Voice Call, Twilio SMS, and Telegram Bot - to the concerned recipients. The notifications include the type of the hazard, GPS location, Google Map URL and the image evidence photo. The Firebase module provides real-time database to the web dashboard. Administrators can easily view current alerts with images, GPS location and a Google Map URL on the web dashboard; additionally, administrators can control the safety devices remotely, for example, the fan and warning light, through the relay module.



SYSTEM WORKFLOW



REQUIREMENT ANALYSIS

Functional Requirements:

- Detect fire incidents using a flame sensor.
- Detect gas leakage and smoke concentration using the MQ-2 gas sensor.
- Capture images of fire incidents using the ESP32-S3 camera module.
- Acquire and transmit real-time GPS coordinates of the incident location.
- Send emergency notifications through multiple communication channels, including Telegram, SMS, voice calls, and email.
- Provide a web-based dashboard for remote monitoring and manual control of safety equipment such as warning lights and exhaust fans.

Non-Functional Requirements

- Provide an emergency response time of less than five seconds.
- Ensure high reliability and fault tolerance during continuous operation.
- Utilize a scalable cloud-based architecture for efficient data processing and communication.
- Maintain the security and privacy of user information.
- Operate with low power consumption to support continuous deployment.
- Support uninterrupted 24/7 environmental monitoring.

LOGIN PAGES

Twilio login

A Twilio account is used to provide SMS and automated voice call notification during emergency situations. The system uses the Twilio API credentials (Account SID and Auth Token) for sending instant text messages and voice calls message to the owner and emergency contacts with information about fire or gas alert and GPS location. This allows instant notification regardless of your internet-based messaging services.

SMS Alerts

Service The SMS alert service is based on the Twilio API and used to send instant emergency text message upon the detection of a fire or gas leakage. An SMS containing the type of emergency, GPS coordinates and the Google location Link is automatically sent to the registered owner and designated emergency contacts. Thus the user is always alerted promptly, especially when they are not connected to the internet for improving response time in emergencies.

Emergency Alert!

Fire detected at:

Latitude: XX.XXXX

|
▼

Longitude: XX.XXXX

|
▼

Immediate action required.

Gmail login

Employing an account, a push of emergency emails can be performed using the application of Google Apps script. The scripting will email all of the users an HTML email to include info such as the kind of danger (fire/gas leak), GPS locational data (with a link to Google maps), and image. All communications are performed via Gmail email to ensure strong, secure communications.

Telegram Bot

Telegram Bot is created using Bot Father and the Bot API Token is added to the system. The registered users subscribe to bot and as soon as the emergency occurs, they are notified over Telegram with the following information: type of emergency, the picture taken, GPS location, and link to Google Maps. Telegram provides real-time notifications to multiple users and it does it fast and reliably.

Firebase Login

We develop a Firebase project for Firebase Realtime Database and Firebase Authentication. Administrator login to the Web dashboard is secured using Firebase Authentication, while Realtime Database synchronizes in real-time sensor data, hazard status, captured images and remote control commands for devices like exhaust fan and warning light.

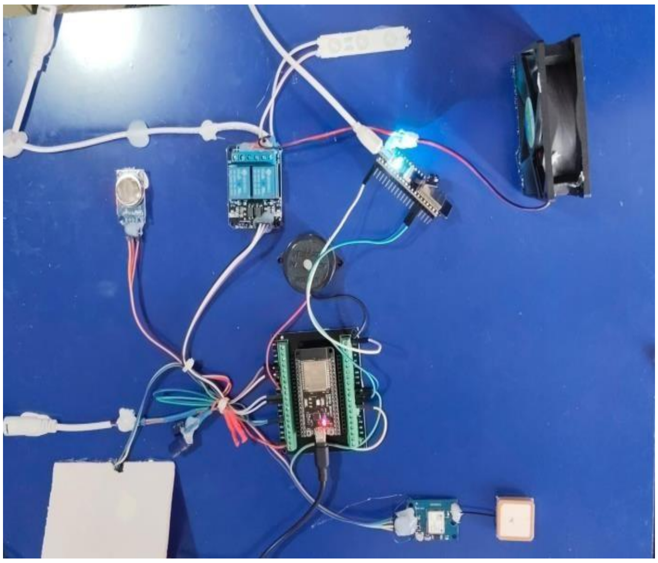
Cloudinary Login

Images taken by the ESP32-S3 Camera Module in emergency situations are stored on a Cloudinary account. Once images are uploaded, Cloudinary provides a secure URL to be attached to email, Telegram and dashboard notifications for users to see proof of the incident

RESULTS

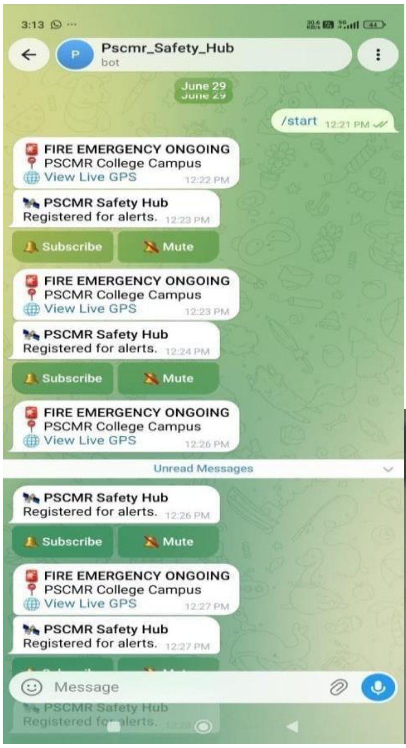
The proposed **Real-Time Fire Image and Location Dispatch Network** successfully detects fire and gas leakage in real time using IoT sensors. The system captures emergency images, retrieves GPS coordinates, and uploads the information to the cloud for live monitoring. Instant notifications are delivered through Telegram, Twilio SMS, voice calls, and email, enabling a rapid emergency response. The web dashboard provides real-time hazard status and allows remote control of safety devices such as the exhaust fan and warning light. Overall, the system demonstrates high reliability, fast response time, and improved safety through integrated edge-cloud technology and multi-channel emergency communication.

Executing with a Screen Shots:

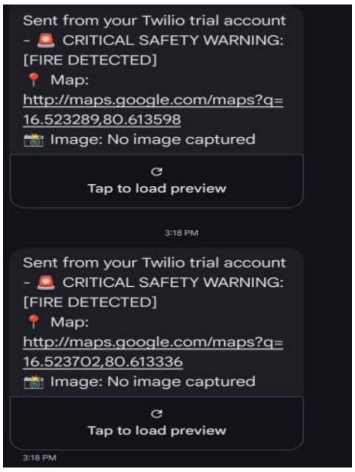


6.2.1 DASHBOARD OF Real-Time Fire Image and Location Dispatch Network utilising Edge Cloud Integration, Twilio, and Telegram.

TELEGRAM BOT NOTIFICATIONS:



SMS NOTIFICATIONS



GMAIL NOTIFICATIONS:

FUTURE SCOPE

The system proposed in this paper is very promising but can be improved with state-of-the-art technologies. Deep learning and artificial intelligence can be employed to detect and identify smoke and fire more accurately and lower the false-alarms. An Android and iOS app can be used to develop a live camera streaming system and device remote control. An updated version of the system can be integrated with local fire stations for automatic fire alarm and providing GPS coordinates along with live images from surrounding cameras and sensors. The system can be improved with addition of other environment sensors such as carbon monoxide sensors and air quality sensors to improve the fire detection and hazard level. Machine learning techniques can be used for analysis of sensor data to predict potential hazards and fire risks. The device can be powered with solar power including a backup battery to make it independent of the electricity network. LoRa and mesh networking can be used to network large number of these devices for industry or smart campus.

Cloud-based technology can be used for generating reports on safety and hazard levels and allow foggy area detection. Autonomous drone-based remote detection system and smart building management system can be employed for better and faster emergency management.

CONCLUSION

A cutting-edge IoT-based emergency alert program Real-Time Fire Image and Location Dispatch Network has been cultivated and launched successfully. The synthesis of sensor-based detection, real-time image processing, GPS tracking and cloud computing into a single system makes this a very sensitive and resilient configuration for dealing with critical situations. In the long testing stage, the prototype was capable of detecting fires and gas leaks with a high degree of reliability, thanks to its sensors, and photographing with the ESP32-S3 camera; because of GPS, it was taking the fire location into account; and using the exact same sensors with the auto-generated alarm, it was dispatching and coordinating an alarm notification to Telegram, email, SMS and by phone. The processing burden was shifted from the microcontroller to the cloud, thus reducing the burden from the microcontroller, which in turn helps to improve communication performance in a critical situation where time is an important factor. Google Apps Script gave the team a cloud-based back-office to trigger simultaneous notifications to multiple platforms and the Firebase dashboard enabled them to easily control external safety appliances by hand, such as exhaust fans and warning LEDs. The final test showed the product to be efficient in delivering rapid and accurate alerts in mock disasters and emergency conditions. To sum up, the realization is a cost-effective and highly scalable solution for real-time monitoring of fire and gas in a safe way. Its modular structure makes it the ideal technology to be installed in any campus, industrial park, warehouse or smart complex, where the precise and quick detection and alerting of a threat could be a matter of life or death.

References

[1] X. L. Yang, Y. C. Li, and Q. Chen," Automated Image-Based Fire Detection and Alarm System Using Edge Computing and Cloud-Based Platform," 2024.

[2] A. Saputra and A. Aswardi," IoT Fire Detector with Telegram Notification," 2024.

[3] D. Kurnia, R. Setiawan, and Janizal," Development of IoT Systems for Fire Detection Tools Using ESP8266 and Telegram Notifications," 2023.

[4] R. S. Kharisma and M. Ibrahim," Digital Image Based IoT Intelligent Fire Detection with Telegram Notification," 2025.

[5] P. Vimala Imogen, R. Rakshana, M. Kaviya, and S. Sharmila, "Real-Time Fire Surveillance with Machine Learning Twilio Integration," 2024.