Internet of Things 27 (2024) 101248

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Towards early forest fire detection and prevention using AI-powered drones and the IoT

Montaser N.A. Ramadan [a], Tasnim Basmaji [b], Abdalla Gad [b], Hasan Hamdan [c], Bekir Tevfik Akgün [d], Mohammed A.H. Ali [a], Mohammad Alkhedher [c], Mohammed Ghazal [b,*]

[a] *Faculty of Engineering, Mechanical Engineering Department, Universiti Malaya, Kuala Lumpur, Malaysia*
[b] *Electrical and Computer Engineering Department, College of Engineering, Abu Dhabi University, Abu Dhabi, United Arab Emirates*
[c] *Mechanical Engineering Department, College of Engineering, Abu Dhabi University, Abu Dhabi, United Arab Emirates*
[d] *Faculty of Computer and Information Sciences, Yeditepe Üniversitesi, Istanbul, Turkey*

A R T I C L E I N F O

Keywords:
Internet of Things (IoT)
LoRaWAN network
Wildfire
Artificial intelligence (AI)
Fire detection
Unmanned aerial vehicle (UAV)
Autonomous navigation

A B S T R A C T

Wildfires are a significant natural hazard, resulting in financial losses, human deaths, and environmental damage. Due to the rising severity and frequency of wildfires, wildfire management and detection have recently received increased attention worldwide. Monitoring potential risk areas and early fire detection are critical factors for shortening the reaction time and reducing the potential damage. Conventional wildfire detection techniques like satellite imaging and remote camera-based sensing need more latency and low reliability. To tackle these limitations, this paper proposes a novel airborne UAV-based IoT (UIoT) system for wildfire sensing, detection, and extinguishing. It presents the design of low-cost and low-maintenance fire-detecting IoT nodes for large-scale deployment. It also proposes, investigates, and reports on several connectivity architectures using the LoRaWAN protocol for UIoT systems. We geolocate forest trees in a terrain-mapping exercise for precise fire localization. We then deploy an autonomous drone with a visual camera that utilizes our novel classification network to detect the presence of fire followed by fire detection and particle filter-based tracking to center fire at the center of the frame. We use a cloud back-end server for monitoring, analysis, and reporting. Our results show that our low-cost and long-range IoT nodes accurately detect fire within 1-5 min after fire ignition. Our fire classification network achieved an accuracy of 99.46% and a mean average precision of 99.64%. Numerical results suggest that the proposed UAV-IoT-based fire classification offers a fast and reliable wildfire recognition and extinguishing solution while minimizing power consumption and increasing the battery lifespan.

1. Introduction

The proposed system was identified per the evidence and noticed statistics of forest and wildfires that occur yearly and internationally due to their negative and significant impact on the environment and society. According to the Congressional Research Service in the United States of America, from the years 2017 to 2021, the number of fires was (in thousands) 71.5 in 2017, 58.1

* Corresponding author.
E-mail addresses: 23063016@siswa.um.edu.my (M.N.A. Ramadan), tasneem.bas@gmail.com (T. Basmaji), abdalla.gad@adu.ac.ae (A. Gad), 1065775@students.adu.ac.ae (H. Hamdan), bekirtevfik.akgun@yeditepe.edu.tr (B.T. Akgün), hashem@um.edu.my (M.A.H. Ali), mohammad.alkhedher@adu.ac.ae (M. Alkhedher), mohammed.ghazal@adu.ac.ae (M. Ghazal).

https://doi.org/10.1016/j.iot.2024.101248
Received 18 February 2024; Received in revised form 25 May 2024; Accepted 6 June 2024
Available online 13 June 2024
2542-6605/

Table 1
The number of human-caused wildfires in different regions in the United States of America.

Year	Alaska	Northwest	Northern California	Southern California	Northern Rockies	Great Basin	Southwest	Rocky Mountains	Eastern Area	Southern Area	Total
2021	262	3,075	3,621	5,083	2,981	1,362	1,657	2,146	10,725	21,729	52,641
2020	181	3,107	4,248	5,295	2,779	2,154	2,743	2,063	13,004	17,989	53,563
2019	349	2,260	3,284,	4,556	1,623	1,387	1,895	948	5,712	22,101	44,115
2018	229	2,787	3,428	4,322	2,137	1,631	2,072	1,394	6,822	26,754	51,576
2017	209	2,150	3,445	5,201	2,646	1,988	2,172	2,150	9,757	33,828	63,546
2016	343	2,082	3,266	3,900	1,784	1,251	2,084	2,048	11,163	33,011	60,932
2015	351	2,898	3,802	3,778	2,246	985	1,327	1,909	11,553	30,067	58,916

Table 2
Burned area in acres of human-caused wildfires in different regions in the United States of America.

Year	Alaska	Northwest	Northern California	Southern California	Northern Rockies	Great Basin	Southwest	Rocky Mountains	Eastern Area	Southern Area	Total
2021	9,894	365,742	1,024,538	82,405	243,603	72,416	351,928	263,758	122,382	487,093	3,023,759
2020	284	1,516,910	1,229,991	927,722	230,928	431,487	368,729	791,996	59,858	440,908	5,998,813
2019	44,061	140,321	190,426	36,146	31,894	126,584	202,918	49,743	38,773	356,188	1,217,324
2018	28,946	609,578	1,404,463	324,102	48,372	1,030,761	309,546	504,416	374	1,329,945	5,640,489
2017	6,890	465,864	343,195	532,640	135,131	788,769	207,818	645,162	41,459	1,663,548	4,830,476
2016	10,069	436,106	93,699	464,718	84,249	333,318	219,103	530,831	97,094	1,497,423	3,766,610
2015	26,652	702,206	308,762	96,990	77,905	49,066	19,596	163,871	100,094	467,319	2,012,461

Table 3
Burned area caused by wildfires in the European, Middle Eastern, and North African regions.

Year	Portugal	Spain	France	Italy	Greece	Total EUMED5	Other countries
2020	67.17	65.923	17.077	55.656	9.3	215.126	72.873
2019	42.084	83.963	23.477	36.034	9.153	194.711	42.502
2018	44.578	25.162	5.124	19.481	15.464	109.809	50.309
2017	540.63	178.234	26.378	137.103	13.393	895.738	78.399
2016	161.522	65.817	16.093	47.926	26.54	317.898	34.602
2015	64.443	103.2	11.16	41.511	7.096	227.41	32.649

in 2018, 50.5 in 2019, 59.0 in 2020, and 59.0 in 2021. The number of acres burned was (in millions) 10.0 in 2017, 8.8 in 2018, 4.7 in 2019, 10.1 in 2020, and 7.1 in 2021 [1]. Tables 1 and 2 show the number of human-caused wildfires and the corresponding burned area throughout the years 2001 to 2020 in the United States of America, taken from the National Interagency Fire Center database. The data show a yearly periodic decrease and increase in the number of fires and the corresponding total burned area.

Similarly, the official European Environmental Agency reported on the burned area (*1000 ha) in different European countries in the technical report titled "Forest Fires in Europe, Middle East, and North Africa 2020". We summarize the report findings in Table 3. As can be seen, the collected data shows the same periodic trend, where the number of fires would start at a lower value in a particular year, increase, and then decrease again. The same trend was noted for years [2]. Other countries listed in the last entry of the table include North African countries, Middle Eastern countries, and other countries in Europe outside of the European Union, such as Turkey and Ukraine.

Wildfires impact the environment, society, and economy of any affected region. This impact includes the economic loss of property and value of the damaged area, loss of humans, animals, and plant population, and the production of air-borne pollutants as a side effect of the burning process of biomass in the affected area of wildfires. As reported by the authors in by [3], the pollution and property losses caused by wildfires in the Californian region are increasing yearly, with a trend similar to the human-caused wildfires, which are overgrowing. Thus, wildfires are caused by many factors, including human-caused wildfires, lightning-caused, weather-caused, and even can be caused by unknown factors. However, the most visible and increasing cause is the human type due to negligence of the laws and regulations. The authors in [4] estimated that the losses in California only show a total of $148.5 billion (US$) in 2018, which included a $27.7 billion loss of capital, $32.2 billion in health costs, and $88.6 billion in indirect losses.

In this research work, we propose a low-cost, low-maintenance, location-aware, and large-scale deployable system that uses the IoT and an AI pipeline for the early detection, tracking, and reporting of forest fires. The proposed system would dispatch a drone autonomously to the detected fire location to extinguish the fire, warn the starters, and alert authorities. We investigate several connectivity architectures, including static gateways connecting neighboring sensing nodes architecture, mobile drone-borne LoRa gateway architecture, and cloud-connected sensing nodes architecture. Our IoT sensing nodes use LoRa modules to transmit data wirelessly, covering a large forest area. They are programmed to enter a deep sleep mode after collecting and transmitting the sensed readings to reduce power consumption. The system dispatches the accompanying drone for precise fire detection and tracking. We propose an AI-based pipeline to detect and track the center of the fire accurately. The drone has a two-way audio channel to alert fire starters and an extinguishing bomb for fire-fighting. The system reports detected fires to authorities to expedite fire containment.

The remainder of the paper is organized as follows: Section 2 presents the literature review of the work and its novelty. Section 3 introduces the materials and methods. Section 4 discusses the testing and validation results of our proposed system and sub-systems. Section 5 presents the work limitations and future work. Finally, conclusions are drawn in Section 6.

2. Literature review

Recently, governments have been applying two main approaches to detect wildfires: (1) satellite imagery and (2) drone-borne sensing. Authors in [5] proposed an early detection system based on governmental satellite imagery and mapping. The satellite map captured was then input into a detection algorithm based on thermal and infrared readings. Similarly, authors in [6] proposed a satellite imagery-based system to be deployed in Russia. The system aims to overcome the scale and remoteness of some wildfires. NASA deployed The same monitoring approach in Southwestern Amazonia rain forests using a global satellite imagery system. The system depends on multisensory reporting of hot spots (fires) in real-time [7]. The utilization of satellite imagery, combined with machine learning algorithms as described in [8,9], facilitates the early detection of wildfires over vast areas, significantly improving the response times of firefighting efforts. Studies such as [10,11] have shown that integrating multi-modal satellite data can significantly enhance the accuracy of fire detection systems. These technologies not only provide broad area coverage but also enable automated fire perimeter mapping, crucial for effective resource deployment and strategic firefighting. This category of technology underscores the scalability and broad-area monitoring capabilities of satellite systems, which are pivotal in managing wildfires effectively and minimizing environmental impact. Satellite technologies thus play a critical role in modern fire management strategies, enhancing both the efficiency and effectiveness of responses to wildfire incidents.

Various methods are often employed to extinguish wildfires after detection. The first approach is aerial suppression, which involves using manned or unmanned airborne forces to drop suppressants on the flames. Several studies have proposed improving the suppressants. Authors in [12] studied and modeled the probability of wildfire suppression by the Initial Attack (IA), in which they mention that the success of the IA accounts for saving most of the losses resulting from the burned area of the wildfires. The model was deployed in the northeastern region of Spain, where they found the lowest chance of IA success in the northwestern mountains and the highest chance gradient in the coastal belt. A study by the authors in [13] shows that a helicopter-based aerial suppression of wildfire could be ineffective due to the high release rate of wildfire heat. The authors noted that a release rate of 10,000 kW can allow the droplets to penetrate the fires effectively. However, if the release rate increased from 12,500 to 23,000 kW, the droplets would only penetrate the wildfire at a certain angle relative to the plume's centerline, rendering the suppression method virtually ineffective. Research done by the authors in [14] indicates the effects of using different chemical enhancers on fire suppressants compared to regular water. The comparative study was done on an experimental setup of a standard laboratory fire on forest fuel in a wind tunnel through a pressurized system mounted above the burning fire. Their comparative tests showed a low coefficient of variation and high reliability, allowing their research to produce a standard experimental setup to compare different suppressant enhancers. The authors in [15] developed another comparison scheme to extinguish uniform and non-uniform fire scenes using a helicopter bucket fire extinguishing system. It was found that pure water, class AB, gel, and class A extinguishing agents, when used to spray 4-layer, 6-layer, and 12-layer wood fires, would show an adequate depth of cooling of 0.36 m, with the performance of cooling agent class A being the highest and pure water having the lowest performance. Based on these findings, they developed a device that can efficiently add the cooling agent to the helicopter bucket. They were applied to extinguish a small area of wildfire near electrical transmission lines successfully.

The integration of Internet of Things (IoT) and sensor-based technologies in forest fire management systems offers sophisticated solutions for real-time monitoring and efficient wildfire response. Papers such as [16,17] highlight how IoT devices equipped with advanced computational capabilities, such as edge computing [18] and LoRa technology [19], enhance real-time data processing and connectivity across extensive networks. These technologies enable a highly responsive and accurate system for detecting and managing wildfires by leveraging real-time data analysis and communication capabilities. Furthermore, the application of IoT in wildfire management not only improves operational efficiencies but also plays a crucial role in reducing environmental impacts through optimized resource utilization and targeted response strategies. These advances significantly contribute to the development of more adaptable, efficient, and scalable wildfire management systems.

The integration of Unmanned Aerial Vehicles (UAVs) and drones in forest fire detection has significantly advanced the capabilities of monitoring and managing wildfires. These technologies underscore their potential in real-time monitoring, dynamic response, and autonomous operations. Drones equipped with infrared and thermal sensors, such as discussed in [20,21], enhance detection accuracy under diverse environmental conditions, proving crucial for effective surveillance and risk assessment. Adaptive path planning algorithms, as explored in [22,23], enable UAVs to dynamically navigate through complex terrains, ensuring comprehensive area coverage and timely detection of fire outbreaks.

Autonomous drone technologies further underscore their efficiency in real-time data processing and fire perimeter mapping, vital for swift firefighting strategies and resource allocation [23,24]. The real-time capabilities of these drones are underpinned by advancements in sensor technology and image processing algorithms that allow for the immediate transmission and analysis of data, facilitating faster decision-making processes. This literature highlights the transformative impact of UAV and drone technologies in forest fire management, underscoring their role in improving response times, operational efficiency, and overall effectiveness in combating wildfires [25].

2.1. Contribution to the field

We define criteria to evaluate and compare state-of-the-art technologies in forest fire detection and management. The criteria include Integration of Technologies, focusing on the novel use of combined technological approaches; Real-time Capabilities, assessing the ability to process and relay information instantaneously; Accuracy and Efficiency, measuring detection precision and operational effectiveness; Scalability and Environmental Impact, evaluating the ability to expand and sustainably deploy technology; and Adaptability to Adverse Conditions, analyzing performance under varied environmental stresses.

Our literature review explored diverse methodologies integrating AI, IoT, and drone technologies, among others, to address these criteria. Our work contributes significantly to this field by introducing a system that integrates advanced AI algorithms with IoT and drone technology, providing high precision, real-time monitoring, and robust performance across different environmental conditions. This research aims to enhance early detection capabilities, optimize response strategies, and significantly reduce the environmental impact of traditional fire management approaches, thereby addressing critical gaps in current wildfire management strategies. Importantly, our AI-powered and IoT-based system operates at a lower power consumption compared to traditional surveillance cameras. It covers a much larger area at a lower cost, as it does not require multiple cameras to cover blind spots. Unlike surveillance cameras that may only detect fires once they have escalated, or satellite imagery, which may miss smaller fires, our approach is designed to capture fires at their initial stages, significantly reducing management costs and environmental impact (see Table 4).

3. Materials and methods

3.1. System overall description

In this work, we propose a novel airborne UAV-based IoT (UIoT) system to detect, alert, and extinguish wildfire ignition in the least possible time and convey the detection result to the cloud. The system is based on low-cost fire-detecting IoT nodes, custom-made to detect smoke in forests with an aesthetic design, and autonomous fire detection and tracking drone. In the setup stage, we geolocate the IoT nodes attached to forest trees in a terrain-mapping exercise to locate and monitor the fire progress precisely. As seen in Fig. 1, each IoT node integrates an air quality sensor connected to a primary control board that deploys the LoRaWAN protocol to transmit the sensed data to a LORA gateway on a timely basis. The gateway then forwards the collected data to a cloud backend server for processing. In the detection stage, the system processes the data obtained from the IoT nodes, extracts the location of the reported fire, and transmits the GPS coordinates to the mission planning software on the Ground Control Station (GCS) that sends the drone to the corresponding location. The drone then applies our proposed AI fire classifier network to detect the presence of fire. This triggers the detector to track the center of the fire using the particle filter algorithm and uses countermeasures to alert fire starters. Results from detection, tracking, and precise localization become accessible from a companion dashboard. The dashboard allows the system operator to monitor, observe, and track the system reports in real-time.

3.2. System architectures

The IoT system proposed is based on a LoRaWAN network composed of a network gateway that forwards messages between multiple LoRa IoT sensing nodes and a LoRaWAN server. The server directs the packets of each sensing node in the network to an application server. We use Amazon Web Services (AWS) for our cloud server needs, providing high scalability, robust security, and real-time capabilities. AWS supports Auto Scaling and Elastic Load Balancing to ensure our system can handle varying loads by automatically adjusting the number of EC2 instances. For security, AWS encrypts our data at rest using AWS KMS and in transit with AWS Certificate Manager, while AWS IAM enforces multi-factor authentication and role-based access control. AWS's compliance with ISO 27001 and SOC 2 standards ensures stringent security practices. Real-time capabilities are supported by AWS Greengrass for low latency processing and Amazon Kinesis and AWS Lambda for real-time data streaming and processing. The gateway is connected to the network server through the conventional TCP/IP SSL network, while the sensing nodes, using LoRa or FSK modulation, use LoRaWAN and communicate with the gateway. The gateway collects data from the different nodes and connects the nodes to the rest of the network. This work investigates several connectivity architectures to maximize system throughput and minimize response time while maintaining battery life.

3.2.1. Static gateways connecting neighboring sensing nodes

Fig. 2 demonstrates the first investigated architecture, where multiple sensing nodes, distributed across the forest, communicate with static gateways. To maximize the battery life, we design the IoT nodes to wake up from a deep-sleep mode once every 5 min, collect readings, send the collected readings to the closest static gateway using LoRaWAN protocol, and then enter the deep-sleep mode again. In this architecture, the LoRa module works as class A. The data collected via the IoT nodes is then forwarded by the LoRa gateways to the cloud network server using the HTTPS protocol and then to an application server that handles data storage, processing, and visualization while preserving the details of the parent node. Sufficient deployment of LoRa gateways is needed to cover large areas in this architecture, where each gateway connects up to 2000 nodes covering an area of 5 million km^2.

Table 4
Comparison of Forest Fire Detection and Management Technologies.

Ref.	Integration of Technologies	Real-time Capabilities	Accuracy and Efficiency	Scalability and Environmental Impact	Fire Detection Technologies	Fire Management Technologies
[20]	Infrared cameras on UAVs	–	High in specific conditions	–	Infrared imaging	–
[16]	IoT with machine learning	Real-time updates (IoT connectivity, LSTM processing)	High precision (98.6% precision in testing)	–	IoT devices, machine learning	Data-driven decision support
[8]	Satellite imagery, machine learning	–	Enhanced early detection	Broad area coverage	Satellite imagery, machine learning algorithms	Large area monitoring
[10]	Multi-modal satellite data	–	–	Large-scale monitoring	Satellite data fusion	–
[11]	Satellite image processing	–	Accurate perimeter mapping	–	Satellite image processing	–
[9]	Deep learning, satellite images	Real-time monitoring	–	–	Deep learning on satellite images	–
[21]	Thermal and optical sensors on UAVs	–	Reduces false positives	–	Thermal and optical sensor integration	–
[26]	UAV-based thermal imaging	Real-time risk assessment	–	–	UAV-based thermal imaging	Real-time risk assessment
[27]	Multi-sensor fusion	–	Optimizes detection	–	Sensor fusion (optical, thermal)	–
[28]	AI-enhanced UAV surveillance	Immediate detection	Enhances surveillance efficiency	–	AI algorithms, UAV surveillance	Immediate response capabilities
[17]	IoT technologies	Real-time data	Efficient management	–	IoT networks, real-time data processing	Efficient wildfire management strategies
[19]	LoRa-enabled IoT devices	–	–	Scalable through LoRa technology	LoRa IoT devices	–
[18]	Edge computing in IoT	Faster wildfire detection	Reduces latency	–	Edge computing, IoT sensors	–
[22]	Adaptive UAV path planning	Real-time tracking	–	–	UAV path planning algorithms	Dynamic monitoring and response
[23]	UAV-based data processing	Real-time processing	Accurate fire mapping	–	UAV-based systems	Real-time perimeter mapping
[24]	Autonomous UAVs	–	–	–	Autonomous UAV technologies	Autonomous operational strategies
[25]	Multi-sensor imagery, deep learning	Real-time forest monitoring	Increased precision and mAP in low visibility	–	Deep learning, multispectral UAV imaging	Real-time monitoring and decision-making
Our Approach	AI, IoT, drones	Real-time monitoring (AI analysis, IoT communication)	High precision (99.46% accuracy)	Highly scalable (low-cost IoT nodes), minimal environmental impact	AI-based image analysis, IoT sensors	Drone deployment for immediate action

3.2.2. Mobile drone-borne lora gateway

In the second architecture, we introduce a mobile drone-borne gateway by mounting the LoRa gateway on a drone that flies over the forest and connects to the sensing nodes when in range. As can be seen in Fig. 3, the sensing nodes communicate with the drone-borne gateway as it flies within 15 km using the LoRaWAN protocol. The gateway then forwards the data to the network server and then to the application server using the HTTPS protocol, where it is managed as explained in 3.2.1. In this architecture, only one gateway is deployed, and the sensing nodes are always awake with the LoRa module activated as class C.

3.2.3. Cloud-connected sensing nodes

The third architecture investigated is based on cloud-connected IoT nodes. As seen in Fig. 4, the IoT sensing nodes in this architecture integrate a 4G GSM module that transmits the air quality sensed data directly to an application server via the HTTPS protocol. The application server then handles data storage, processing, and visualization.

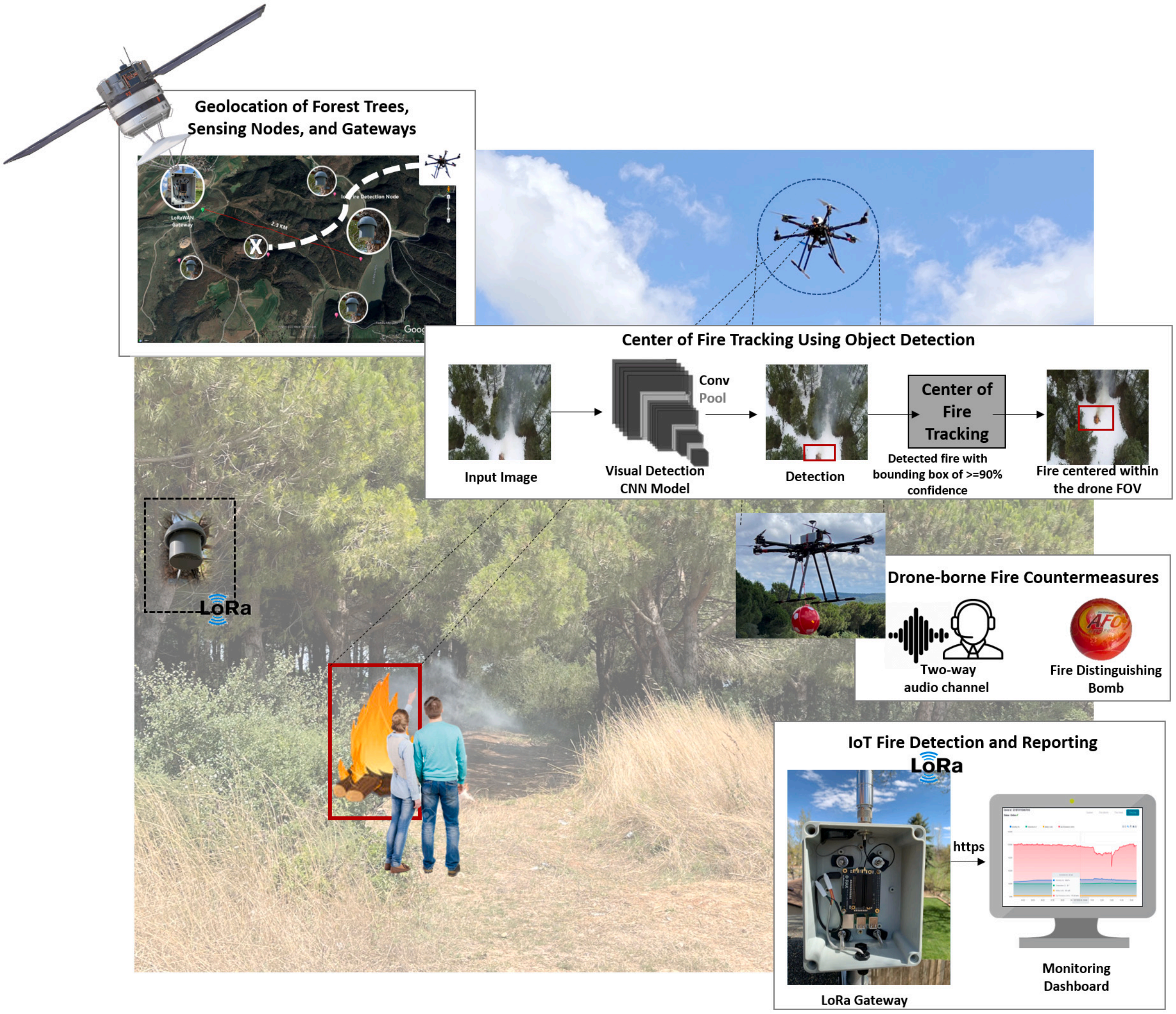

Fig. 1. System overview diagram.

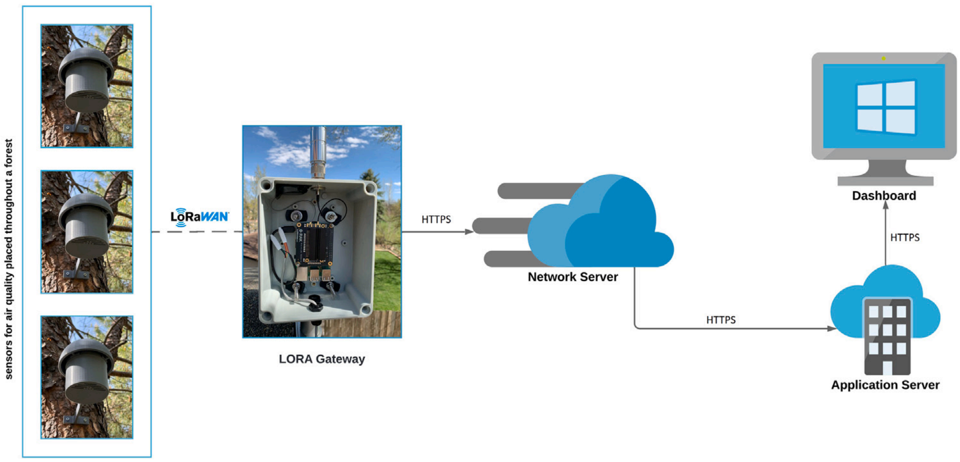

Fig. 2. Static gateways connecting neighboring sensing nodes architecture.

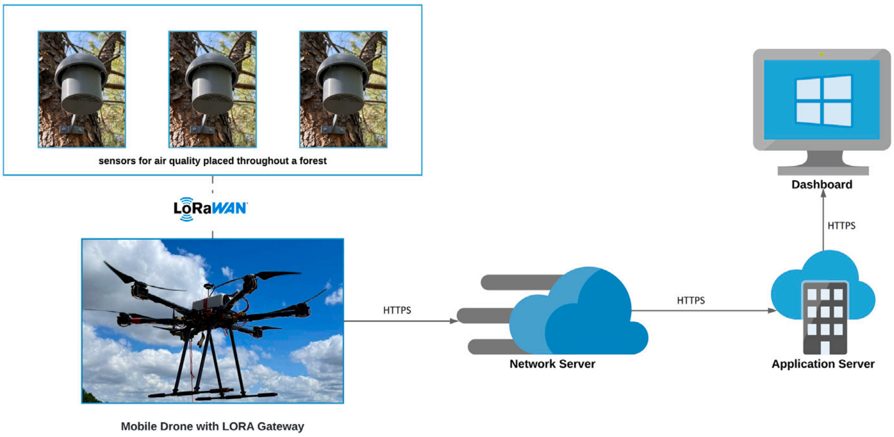

Fig. 3. Mobile drone-borne LoRa gateway architecture.

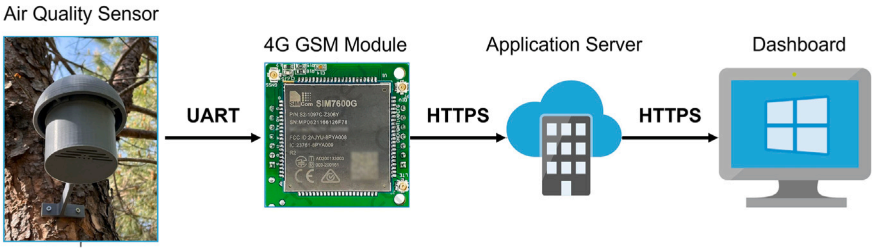

Fig. 4. Cloud-connected sensing nodes architecture.

Table 5
Investigated connectivity architectures compared in terms of advantages, disadvantages, capital cost, running cost, response time, and battery lifespan.

Connectivity Architecture	Advantages	Disadvantages	Capital Cost	Running Cost/month	Response Time	Battery Lifespan
Static Gateways Connecting Neighboring Sensing Nodes	* Low-cost * Large coverage area * Low-latency data collection * Weatherproof	* Gateway requires external power source or solar power solution	**Node:** $40 × 9 = $360 **Gateway:** $200 **Total** = $560	**Servers:** $30	Fast response time (1–5 min)	10 Years
Mobile Drone-borne LoRa Gateway	* Single mobile gateway * Large coverage area	* Slow response time (depends on drone survey) * Time constraints on drone flights * Dedicated base stations are required * Affected by weather conditions	**Node:** $40 × 9 = $360 **Gateway:** $200 **Drone:** $1000 **Total** = $1560	**Servers:** $30 **Battery replacement:** $3.33 per node	Depends on drone's speed and covered area	6 Months
Cloud-connected Sensing Nodes	* Standalone system without gateway * Large coverage area	* Weak GSM signal in forest areas * High-latency data collection * Monthly GSM subscription	**Node:** $70 × 9 = $630	**Servers:** $30 **Battery replacement:** $0.83 per node **GSM subscription:** $45	Fast response time (1–5 min)	2 Years

We compare our system's three investigated connectivity architectures in Table 5, assuming an area of 22,500 km^2. For testing purposes, we deploy nine sensing nodes and a single gateway to cover the set area, where the distance between each consecutive sensing node is 50 m.

By comparing the three connectivity architectures, we opted to deploy the Static Gateways Connecting Neighboring Sensing Nodes architecture due to (1) its scalability with a minimal cost which enables seamless integration of a large number of devices, accommodating future expansion without significant infrastructure modifications, (2) the decentralized nature of the architecture enhances reliability by decreasing the latency in data collection and minimizing single points of failure, and (3)

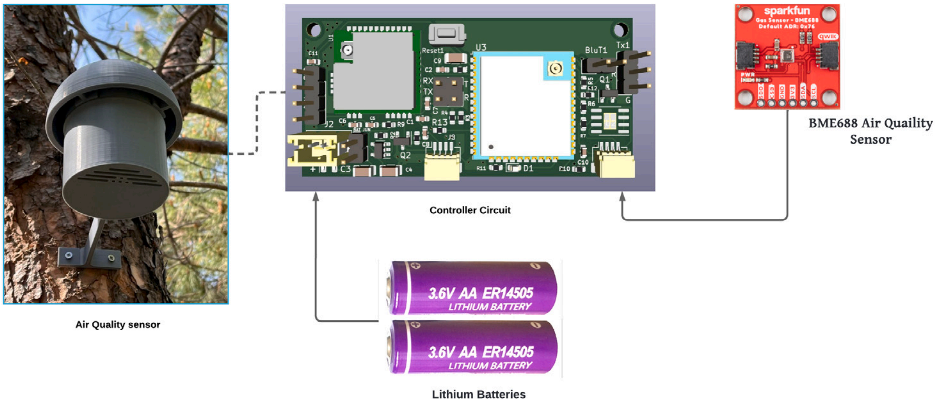

Fig. 5. Hardware connections of the proposed IoT sensing node.

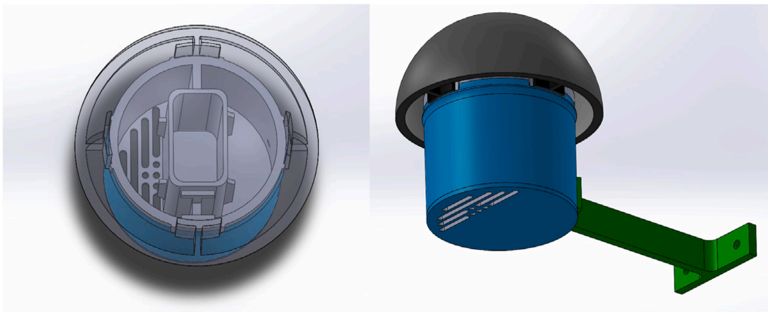

Fig. 6. Left: 3D enclosure compartments and internal design. Right: 3D enclosure lid and external design.

its weatherproof design and communication mechanism ensures resilience against environmental factors such as rain, snow, and extreme temperatures, thereby ensuring uninterrupted operation.

3.3. Design of the IoT fire sensing node

Based on the selected connectivity architecture in Section 3.2, we design our IoT sensing node to integrate a primary control board and an air quality sensor powered using LiPo batteries. The node encapsulates the RAK3172 LoRaWAN module for long-range wireless data transmissions with low power consumption. The BME688 air quality sensor for detecting the presence of fire. It integrates the temperature, humidity, and barometric pressure measurements. The ESP32 Microcontroller with the deep sleep, light sleep, and modem sleep. ESP32 is used to integrate the air quality sensors with the LoRaWAN module. Each node is programmed to send the sensed data once every five minutes and then goes into the deep-sleep mode to preserve power. As illustrated in Fig. 5, the main controller circuit is a low-power, 32-bit microcontroller unit that integrates a long-range transceiver module for LoRaWAN and LoRa applications. In the deep-sleep mode, the main CPU and memory of the microcontroller unit turn off, reducing the power consumption to approximately 3 μA. As the memory gets disabled in the deep-sleep mode, the pre-sensed data gets erased. This sleep pattern is called the ULP sensor-monitored pattern. For air quality sensing and smoke detection, we use an air quality sensor module that measures temperature and humidity levels with high precision and gasses, including VOCs, VSCs, carbon monoxide, and hydrogen at parts per billion (ppb) levels. The hardware connections of the proposed sensing node are illustrated in Fig. 5.

3.3.1. 3D design of the enclosure

We design a light but sturdy 3D model to enclose the proposed IoT sensing node and protect it against water, weather, and sunlight. As seen in Fig. 6, the 3D enclosure is designed to have a dome-shaped lid with several distinct areas in the body that perfectly house the system components. We created holes below the lid and at the bottom of the air quality sensor region, allowing smoke or air to travel through and out of the dome to ensure air circulation. The 3D enclosure is designed to be placed and fixed at a distance from a static surface, i.e., trees.

3.3.2. PCB design of the primary control board

In order to fully integrate the system while reducing power consumption, we design, model, and fabricate a 2-layer printed circuit board (PCB) that hosts the primary control board's IC and other off-chip components. We decrease the size of the PCB to the

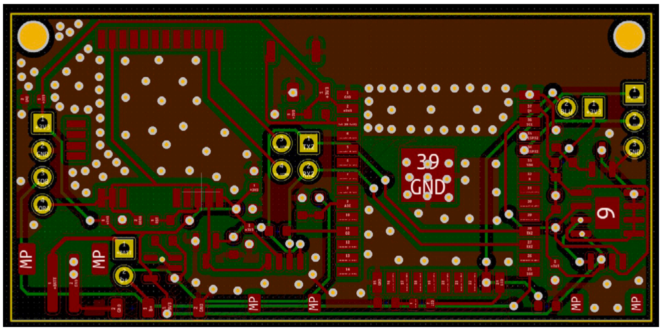

Fig. 7. Proposed PCB design of the primary control board.

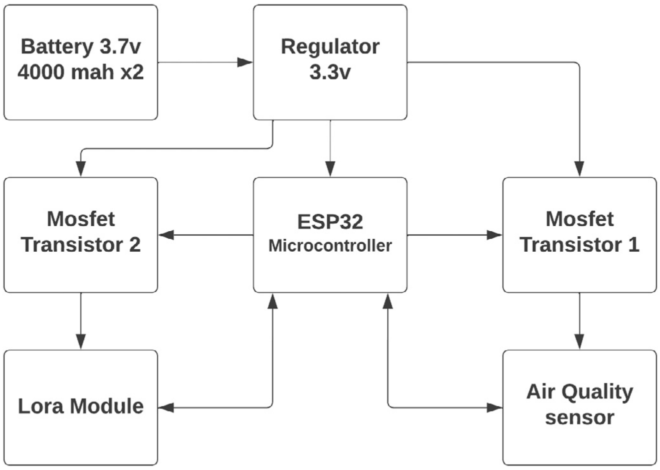

Fig. 8. Proposed power management algorithm connections diagram for low power consumption.

minimum by reducing the length of the tracks for power management purposes. The design, depicted in Fig. 7, considers eliminating disruptive noise interference.

3.3.3. Power management

To reduce the system's power consumption and preserve battery life, we use two MOSFET transistors that completely turn off parts of the system during the deep-sleep mode. To implement this functionality, we follow the connection diagram depicted in Fig. 8, where we program the MCU to send a signal to the MOSFETs before going into the deep-sleep mode. This powers off the LoRa module and the air quality sensor, resulting in deficient power consumption during the deep-sleep mode. When the MCU wakes up, it activates both transistors, powering the air quality sensor and the LoRa module to function. The proposed power management algorithm ensures all main components on the boards are turned off during the deep-sleep mode, leaving only the ULP co-processor to work with an internal real-time clock to calculate the sleeping time.

To evaluate the power consumption of the IoT sensing nodes, we measure the current consumed by the system during the wake-up and the deep-sleep modes. The system operates in the wake-up mode for 5 s, during which the microcontroller activates the air quality sensor and the LoRa module to collect and transmit the data to a LoRa gateway. Once the data is sent, the system goes into deep sleep mode for 5 min, consuming 8 μA. We detail the nominal absorption of each component in the sensing circuit during both the wake-up mode and the deep-sleep mode in Tables 6 and 7, respectively. Eq. (1) is used to calculate the battery life, which is rounded to 4.75 years using our power management configuration.

$$Battery\ Life = \frac{Battery\ Capacity(mah)}{Load\ Current(mah)} \tag{1}$$

$$Sleep\ mode = 5\text{ min} \longrightarrow Consumes\ 8\ \mu\text{A}$$

$$Running\ mode = 5\text{ s} \longrightarrow Consumes\ 14\text{ mA}$$

$$Power\ Consumption\ (\text{mA/h}) = \frac{5 \times 10 \times 14\text{ mA} + 0.008\text{ mA} \times 5 \times 60 \times 10}{3600} = 0.19\text{ mAh}$$

Table 6
Power consumption of the system during wake-up mode.

Part	Power consumption	# of parts	Total
MCU	9 mA	1	9 mA
LoRa Module	3 mA	1	3 mA
Air Quality Sensor	1 mA	1	1 mA
MOSFET Transistor	0.5 mA	2	1 mA
Voltage Regulator	3 µA	1	3 µA
Other components (SMD resistors and capacitors)	2 µA		2 µA
Total power consumption in mA			**14.5 mA**

Table 7
Power consumption of the system during deep-sleep mode.

Part	Power consumption	# of parts	Total
MCU	4 µA	1	4 µA
LoRa Module	0	1	0
Air Quality Sensor	0	1	0
MOSFET Transistor	200 nA	2	400 nA
Voltage Regulator	3 µA	1	3 µA
Other components (SMD resistors and capacitors)	1 µA		1 µA
Total power consumption in µA			**8.4 µA**

$$Battery\ Life = \frac{8000\ \mathrm{mAh}}{0.19\ \mathrm{mAh}} = 42,105\ \mathrm{h}\ and\ 15\ \mathrm{min}$$

∴ Battery Life in Years = 4 years, 9 months, 3 weeks, 2 days, 12 h, and 15 min

3.3.4. IoT system state machine

Fig. 9 illustrates the IoT node process description and the corresponding transitions of the proposed fire detection and intervention system. As seen in the state chart diagram, the system's initial state is the **Idle** state, where the system is fully active and is set to automatically collect sensor readings in the **getSensorReading** state. We set a predefined idle threshold for each sensor as a comparison set point. If the sensed reading exceeds the predefined threshold, the system transitions to the **safe** state. The **fireDetected** flag is set to false in this state. Since no fire has been detected, the system goes into the deep-sleep mode for 5 min through the **sleepSafely** state, then wakes up and goes back to the **getSensorReading** state. If the sensed reading exceeds the threshold, the system transitions to the **danger** state and sets the **fireDetected** flag to true. When a fire is detected, the deep-sleep mode duration decreases to 1 min through the **sleepCautiously** state. In all cases, the system is designed to connect to the LoRa gateway and report on the environmental status as it collects sensor readings. Accordingly, if the air quality sensed readings exceed the normal at a point in time, the system decreases the deep-sleep mode duration. It consumes more power until the air quality returns to normal.

3.4. Autonomous deployment of intervention drone

3.4.1. Geo-mapping of gateways and sensing nodes

IoT nodes in the forest are positioned using a GPS Real-Time Kinematic (RTK) board that relies on the u-blox NEO-M8P-2 module. When using RTK technology, the horizontal position precision is 0.025 m. The distribution of the proposed sensing IoT nodes in the forest is to be determined based on the location of camping spots. After conducting several range tests, the maximum connection range between a sensing node and a gateway in a forest is measured to be 2.3 km, after which the connection becomes unreliable. We deploy five sensing nodes and a single gateway covering an area of approximately 5 km^2 for testing purposes. We mount each sensing node on a tree trunk three meters above the ground and connect the gateway to a distribution line. Then, we determine the precise longitude and latitude of the sensing nodes and the gateway using GPS to geo-map the sensed readings.

In the system setup stage, we develop a geolocation lookup table, shown in Table 8, that maps the unique identifiers of the sensing nodes to the location of the nodes. Once a fire is detected, we use the unique identifier of the sensing node received at the cloud server and map it to determine the location of the detected fire. While geolocation mapping generally has limitations in pinpointing precise locations, our approach mitigates this by utilizing UAVs equipped with a vision-based AI system for exact fire detection and tracking. This combination enhances the accuracy and effectiveness of localizing the exact fire location.

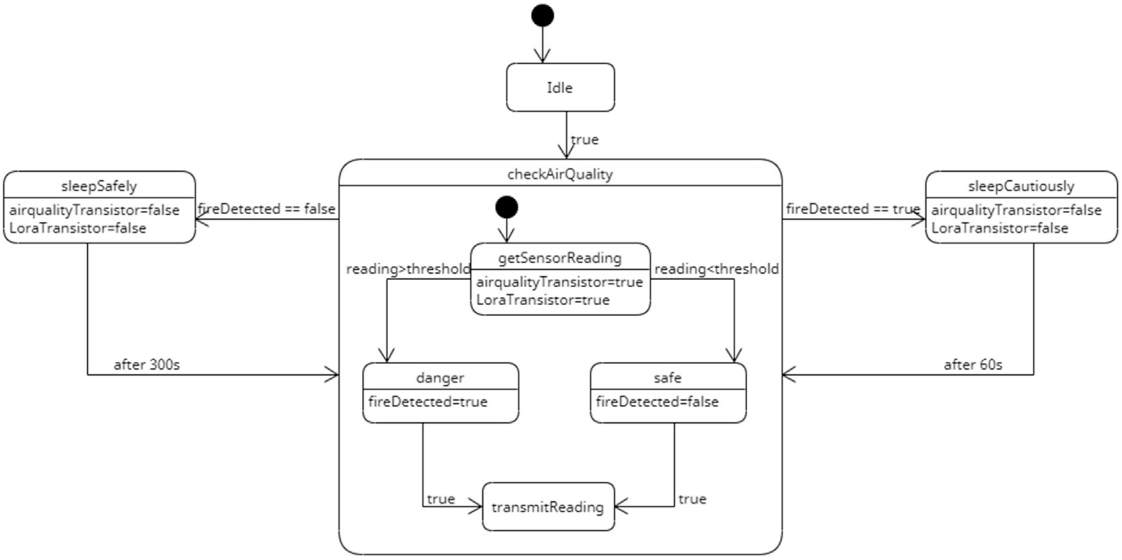

Fig. 9. IoT sensing node statechart.

Table 8
GPS coordinates Lookup table of the deployed IoT sensing nodes and gateway.

Device	Latitude	Longitude	Distance from Gateway	Device EUI
Fire detection node #1	41° 8'26.42"N	28° 43'39.23"E	1.16 km	5E9F1E0857EF65F2
Fire detection node #2	41° 8'23.01"N	28° 42'52.23"E	0.8 km	2f70C925E932F42F
Fire detection node #3	41° 8'59.39"N	28° 44'19.02"E	1.9 km	9E90C1E0857DF259
Fire detection node #4	41° 8'25.23"N	28° 44'30.10"E	2.3 km	5E9E0E0357CF25FC
Fire detection node #5	41° 8'2.12"N	28° 44'14.26"E	2.27 km	4E9C4E0557DF25F3
Gateway	41° 8'47.99"N	28° 42'58.60"E	N/A	2E9C9E0257DF25F2

3.4.2. Mission planning and flight control

We integrate our application server, where the locations and IDs of all sensing nodes are stored, with open-source mission planning software. The integration is intended to ensure the drone gets dispatched automatically to the detected fire location based on the readings obtained from the sensing nodes. To avoid dispatching the drone on false positive alarms, the application server sends the location of the detected fire to the mission planning software only after receiving three consecutive warnings for a particular node. The drone gets dispatched and flies autonomously following the flight path provided by the mission planning software to the identified location, as shown in Fig. 10.

3.4.3. Drone-borne fire countermeasures

Once our autonomous drone reaches the designated site, it initiates the pipeline described in Section 3.5 to localize the fire precisely. If the fire is detected, the drone activates a two-way audio and alert channel, connecting the system operator and the fire-starters to warn and guide them through the extinguishment process. As indicated in Fig. 11, the drone is also equipped with a fire-extinguishing bomb that may be launched if the fire is at its initial stage.

3.5. Fire tracking using object detection

We propose a novel fire detection and tracking system that uses deep learning and computer vision to detect and track forest fires. Our approach utilizes The Raspberry Pi HQ 12-megapixel high-resolution Camera Module fitted on the drone using a 2-axis 3D printed gimbal, as shown in Fig. 12

In deep learning, many large datasets are available for researchers to train their models. However, until recently, a UAV dataset for forest fire detection was lacking. This work uses the publicly available FLAME dataset (Fire Luminosity Airborne-based Machine Learning Evaluation) published in 2021 [29] to train and evaluate our proposed classifier. The FLAME dataset contains aerial images and raw heat-map footage captured by visible spectrum and thermal cameras onboard a drone. It consists of four types of videos: a normal spectrum, white-hot, fusion, and green-hot palettes. In this paper, we focus on RGB aerial images. The dataset contains 48,010 RGB images, split into 30,155 Fire images and 17,855 Non-Fire images for wildfire classification tasks. We use 5615 unique Fire and Non-Fire images to train, test, and validate our CNN-based classifier. Images used have a resolution of 254 × 254 pixels. Fig. 13 represents some samples of the FLAME dataset.

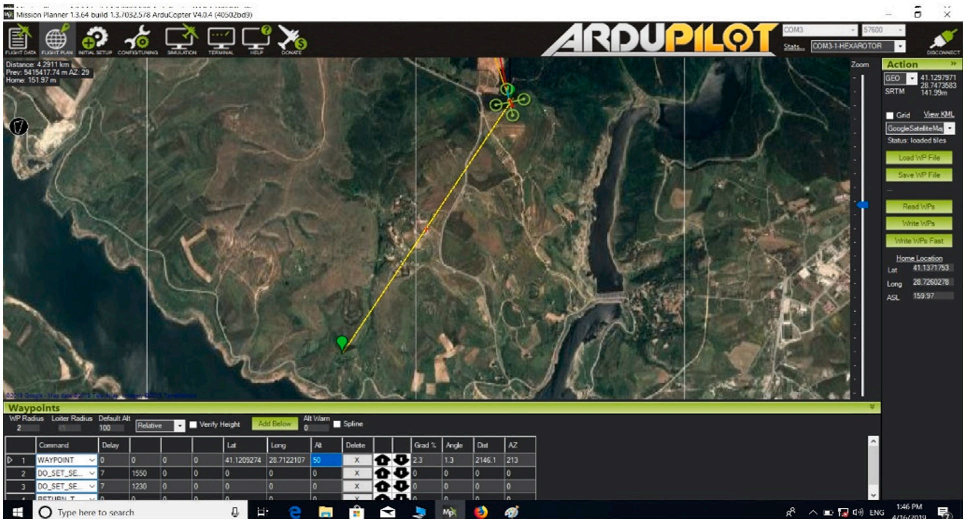

Fig. 10. The autonomous drone flying towards the detected fire location determined by the server.

(a) (b) (c)

Fig. 11. The autonomous drone (a) carrying the fire extinguishing bomb, (b) triggering the release, and (c) reaching the fire on the ground.

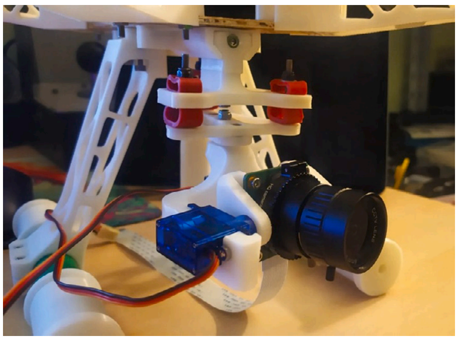

Fig. 12. 3D printed gimbal with HQ raspberry pi camera.

Table 9
Composition of the fire classification dataset.

Class	Data size
Fire	2,854
Non-Fire	2,761

The testing set consists of 561 images with no annotations. The images contain similar characteristics to the training set. Some, but not all, of the locations captured in the testing set are also included in the training set. The images from training and testing datasets are almost equally balanced between the two classes, Fire and Non-Fire, as detailed in Table 9.

We use deep learning to estimate the function f that maps our input visual features or image $I[n]$ to the desired discrete output $y[n]$. Each $y[n]$ label is selected from two possible outcomes (i.e., fire or no-fire). In this research, we do not rely on a predefined

Fig. 13. Composition of the Flame dataset. Left: Fire class sample image. Right: NoFire class sample image.

convolutional neural network model but use Neural Architecture Search (NAS) to design a model that maximizes the accuracy performance criterion. NAS is a subfield of automated machine learning (AutoML) aimed at optimizing neural network architectures autonomously, first introduced in [30]. NAS involves defining a search space of potential architectures, employing strategies like reinforcement learning, evolutionary algorithms, or gradient-based methods to explore this space, and evaluating the performance of each candidate architecture. The goal is to identify the optimal architecture for a specific task without human intervention. Our search strategies include Sequential, Multi-Layer Perceptron (MLP), 2D convolution layers, 2D max-pooling layers, and dropout layers. We use Bayesian optimization to find the most optimal architecture with the highest accuracy. We use

$$f^*, w^* = \underset{f,w}{\operatorname{argmin}} \left[\sum_{I[n],y[n]\in T} L(f(I[n];w), y[n]) \right], \tag{2}$$

where T is the set of training image data taken as a randomly selected 80% of all available data, we do not only aim to find the optimal weight but also the optimal model architecture (e.g., number of layers, their types, their connections, and their hyperparameters). Using our data, we use a recurrent neural network (RNN) as the controller of the fire classification model. The reward signal to train the RNN is based on the validation set D accuracy, which is taken as 10% of the data, with the remaining 10% for testing and performance evaluation. To find the optimal weight and model architecture, two networks are at play: a controller RNN network and a child convolutional neural network. The convolutional neural network produced is then used to find the probability of fire classification, or

$$p^+[n], p^-[n] = f^*(I[n]; w^*). \tag{3}$$

Table 10 details the architecture of the fire classification model. The input layer of this model takes an RGB color image with a size of 224 × 224 pixels. The unique feature of our architecture is that it employs a combination of standard Conv2D and depthwise separable convolutions, structured into multiple Conv blocks that progressively increase the abstraction of features. This hybrid approach leverages the strengths of depthwise separable convolutions in reducing the number of parameters and computational load, while Conv2D layers ensure robust feature extraction. Additionally, the use of multiple custom Conv blocks allows for hierarchical feature representation, capturing both spatial and cross-channel interactions at different scales. These architectural choices collectively enhance the model's capability to distinguish fire from non-fire scenarios with greater accuracy and resilience to variations in image quality and environmental conditions. In Table 10, it can be observed that each input image undergoes several convolution layers starting with a 48 convolution kernel of 3 × 3 with the stride of 2 in the first layer. Further, convolutional blocks are organized as three-layer bottleneck blocks with several filters with the stride of 1. At the bottom of the network, a fully connected layer is inserted to perform drone classification, where each unit in the last layer is linked to 2 output probabilities by utilizing the Softmax function. Accordingly, the proposed classifier produces a probability of positive class prediction, or "Fire", and negative class prediction, or "Non-Fire", represented by $p^+[n]$ and $p^-[n]$, respectively.

We apply the test-set cross-validation technique, where we opted to use the combination of 80:20 for the ratio of training to the testing dataset, as described in Table 11, in which the training, testing, and validation sets used have 4493, 561 and 561 images, respectively.

The fire detection and localization algorithm is initialized once the presence of fire is determined by the classification network. When assessing the performance of estimation and tracking methods, such as extended Kalman filters (EKFs), unscented Kalman filters (UKFs), and particle filters (PFs), as explored in the literature, the choice depends on the specific context, with particle filters demonstrating high accuracy in handling nonlinearities and uncertainties. In [31], the authors evaluate the performance of these filters in estimation tasks. They find that while EKFs and UKFs provide efficient approximations to the true state distribution, they may suffer from inaccuracies when dealing with nonlinear or non-Gaussian systems. On the other hand, particle filters, as demonstrated in [32], offer a more flexible approach, particularly suitable for handling nonlinear and non-Gaussian problems. However, they may suffer from high computational complexity, especially in high-dimensional state spaces. In [33], the authors compare particle filters with range-parameterized and modified polar EKFs specifically for angle-only tracking scenarios. They find

Table 10
Description of the proposed CNN-based fire classifier.

Layer name	Input size	Output size	Specifications (Kernel Size, No. of Kernels, Stride)
Conv2D	224 × 224 × 3	112 × 112 × 48	3 × 3, 3, 2
DepthwiseConv2D	112 × 112 × 48	112 × 112 × 48	3 × 3, 48, 1
Conv2D	112 × 112 × 48	112 × 112 × 24	1 × 1, 48, 1
Conv2D	112 × 112 × 24	112 × 112 × 72	1 × 1, 24, 1
DepthwiseConv2D	112 × 112 × 72	56 × 56 × 72	3 × 3, 72, 2
Conv2D	56 × 56 × 72	56 × 56 × 24	1 × 1, 72, 1
Conv1_Block	56 × 56 × 24 56 × 56 × 72 56 × 56 × 72	56 × 56 × 72 56 × 56 × 72 56 × 56 × 24	1 × 1, 24, 1 3 × 3, 72, 1 1 × 1, 72, 1
Conv1_Block	56 × 56 × 24 56 × 56 × 72 56 × 56 × 72	56 × 56 × 72 56 × 56 × 72 56 × 56 × 24	1 × 1, 24, 1 3 × 3, 72, 1 1 × 1, 72, 1
Conv2D	56 × 56 × 24	56 × 56 × 72	1 × 1, 24, 1
DepthwiseConv2D	56 × 56 × 72	28 × 28 × 72	5 × 5, 72, 2
Conv2D	28 × 28 × 72	28 × 28 × 56	1 × 1, 72, 1
Conv2_Block	28 × 28 × 56 28 × 28 × 168 28 × 28 × 168	28 × 28 × 168 28 × 28 × 168 28 × 28 × 56	1 × 1, 56, 1 5 × 5, 168, 1 1 × 1, 168, 1
Conv2_Block	28 × 28 × 56 28 × 28 × 168 28 × 28 × 168	28 × 28 × 168 28 × 28 × 168 28 × 28 × 56	1 × 1, 56, 1 5 × 5, 168, 1 1 × 1, 168, 1
Conv2D	28 × 28 × 56	28 × 28 × 336	1 × 1, 56, 1
DepthwiseConv2D	28 × 28 × 336	14 × 14 × 336	5 × 5, 336, 2
Conv2D	14 × 14 × 336	14 × 14 × 104	1 × 1, 336, 1
Conv3_Block	14 × 14 × 104 14 × 14 × 624 14 × 14 × 624	14 × 14 × 624 14 × 14 × 624 14 × 14 × 104	1 × 1, 104, 1 5 × 5, 624, 1 1 × 1, 624, 1
Conv3_Block	14 × 14 × 104 14 × 14 × 624 14 × 14 × 624	14 × 14 × 624 14 × 14 × 624 14 × 14 × 104	1 × 1, 104, 1 5 × 5, 624, 1 1 × 1, 624, 1
Conv2D	14 × 14 × 104	14 × 14 × 624	1 × 1, 104, 1
DepthwiseConv2D	14 × 14 × 624	14 × 14 × 624	3 × 3, 624, 1
Conv2D	14 × 14 × 624	14 × 14 × 136	1 × 1, 624, 1
Conv4_Block	14 × 14 × 136 14 × 14 × 816 14 × 14 × 816	14 × 14 × 816 14 × 14 × 816 14 × 14 × 136	1 × 1, 136, 1 3 × 3, 816, 1 1 × 1, 816, 1
Conv2D	14 × 14 × 136	14 × 14 × 816	1 × 1, 136, 1
DepthwiseConv2D	14 × 14 × 816	7 × 7 × 816	5 × 5, 816, 2
Conv2D	7 × 7 × 816	7 × 7 × 272	1 × 1, 816, 1
Conv5_Block	7 × 7 × 272 7 × 7 × 1632 7 × 7 × 1632	7 × 7 × 1632 7 × 7 × 1632 7 × 7 × 272	1 × 1, 272, 1 5 × 5, 1632, 1 1 × 1, 1632, 1
Conv5_Block	7 × 7 × 272 7 × 7 × 1632 7 × 7 × 1632	7 × 7 × 1632 7 × 7 × 1632 7 × 7 × 272	1 × 1, 272, 1 5 × 5, 1632, 1 1 × 1, 1632, 1
Conv5_Block	7 × 7 × 272 7 × 7 × 1632 7 × 7 × 1632	7 × 7 × 1632 7 × 7 × 1632 7 × 7 × 272	1 × 1, 272, 1 5 × 5, 1632, 1 1 × 1, 1632, 1
Conv2D	7 × 7 × 272	7 × 7 × 1632	1 × 1, 272, 1
DepthwiseConv2D	7 × 7 × 1632	7 × 7 × 1632	3 × 3, 1632, 1
Conv2D	7 × 7 × 1632	7 × 7 × 448	1 × 1, 1632, 1

(*continued on next page*)

Table 10 (*continued*).

Layer name	Input size	Output size	Specifications (Kernel Size, No. of Kernels, Stride)
Conv2D	7 × 7 × 448	7 × 7 × 1280	1 × 1, 448, 1
Mean	7 × 7 × 1280	1280	–
FullyConnected	1280	2	–
Softmax	2	2	–

Table 11
Data distribution.

Data type	Training	Validation	Testing
Percentage	80%	20%	
		10%	10%

that while particle filters exhibit superior performance in handling nonlinearities and uncertainties, the choice between them and EKF variants may depend on the specific tracking context and computational constraints. In our fire detection system, upon a positive class detection, we utilize the particle filter to track the detected fire particles within a frame. The Particle filter, also known as the condensation filter, implements a Monte Carlo and recursive Bayesian estimation approach. It operates by representing the posterior probability density function using a set of particles, each serving as a discrete sample that embodies a hypothesis of the state. These particles are randomly drawn from the prior density, accommodating noisy and partial observations. Notably, the particle filter excels in handling nonlinear state–space models, with the flexibility to adapt to various initial state and noise distributions. Its core components encompass an algorithm and an observation model. We use the particle filter to track the detected fire particles within a frame in case of a positive class detection. The Particle filter, or condensation filter, is a method based on Monte Carlo and recursive Bayesian estimation. Given noisy and partial observations, it uses a set of particles to represent the posterior probability density function. Each particle is a discrete sample that represents a hypothesis of the state and is randomly drawn from the prior density. The state–space model can be nonlinear, and the initial state and noise distributions can take any form required. The particle filter includes two essential parts: an algorithm and observation models as described in Appendix.

4. Validation and results

This section presents a detailed discussion of the results for each of the introduced system components, including the sensing accuracy of the IoT nodes, power consumption, system connectivity, response time, drone deployment, and fire detection and tracking.

4.1. Experimental setup

We conduct three testing scenarios to validate the accuracy of the sensing nodes in detecting the presence of fire. The three tests include an idle case where there is no fire, paper burning where the fire is manageable and no interference is required, and a barbecue where external interference is required to manage the fire before it gets out of control. The level of CO_2 detected by the sensing node determines the severity of the fire. The connectivity range between the sensing nodes and the LoRaWAN gateway is determined by placing multiple sensing nodes at a measured distance from the gateway. This specifies the maximum range at which the data transmission goes smoothly without interruptions.

To determine the response time of our fire sensing nodes, we mount one to a tree in the forest and start a barbecue. Utilizing this setup, we measure the time it takes for the sensing nodes to send an alarm for the presence of a fire. In addition, we measured the flight time for our drone to reach the fire location to extinguish the fire. This test has been conducted 10 times to accurately evaluate the overall response time for our approach. Finally, we evaluate our fire detection and tracking model using the FLAME dataset regarding accuracy, precision, recall, and the F1-score. We also compare our model with state-of-the-art fire classification models that have been tested on the same dataset to validate the performance superiority of our architecture.

4.2. Sensors accuracy

To validate the accuracy of the readings collected through the proposed IoT sensing nodes, we compare the gas measurements obtained in different scenarios on-site to the values measured using the industrial level multi-gas detector, BH-4S. This detector has an accuracy of ≤+−5% and a response time of ≤30 s. We detail four testing scenarios in Table 12. As can be seen, the IoT sensing node was able to accurately detect the CO2 gas level and report fire accordingly. By comparing the difference between the detected and the actual CO2 readings, the IoT sensing node achieved an accuracy of 99.2%. These findings support the validity of the proposed sensing nodes.

Table 12
Detected and measured CO_2 values in four different scenarios on-site.

Testing #	Testing Scenario	Detected CO2 Level (ppm)	Measured CO2 Level (ppm)	Fire detected
1	Idle	24.0	22.88	No
2	Burning paper	246.0	243.84	No
3	Barbecue	971.0	968.79	Yes
System Accuracy			**99.2%**	

Table 13
Testing scenarios conducted on-site to evaluate the response time of the system.

Testing #	Testing scenario	Distance from closest sensing node	Wind speed	Wind direction	System response time
1	Barbecue	20 m	8 km/h	NE	5 min
2	Barbecue	50 m	19 km/h	N	4 min
3	Barbecue	5 m	12 km/h	NE	3 min
4	Barbecue	30 m	21 km/h	NE	3.5 min

Fig. 14. The map of the forest showing the locations set for the testing nodes and gateway.

4.3. Connectivity testing

To determine the connectivity range of the selected architecture, we conducted several range tests by deploying five sensing nodes and a single gateway. We started our testing by placing the sensing nodes at a distance of 3 km from the gateway and then decreasing the distance by 50 m after each run. Based on the connectivity and transmission results, the maximum connection range between a sensing node and a gateway in a forest was measured to be 2.3 km, after which the connection becomes unreliable. Fig. 14 shows the locations where we set the testing gateway and sensing nodes and the distance measured between the nodes and the gateway.

4.4. Response time

To test the system's response time, we measured the time taken from when we started a fire in the forest until the drone took off. Based on our testing results, the system can detect nearly every type of fire and send off the drone for accurate detection, monitoring, alerting, and tracking under 5 min in ideal conditions. The response time may decrease or increase slightly based on the wind speed and direction affecting the fire's forward rate. Table 13 details the conducted testing scenarios. Once the drone takes off, it flies at a 70 km/h speed until it reaches the detection location.

We further carried out multiple tests on the IoT sensing node to determine the response time for fire detection. Table 14 displays various experiment results, which illustrate fluctuations in response time. This variability is attributed to the IoT node's operational cycle, which involves 5 s in active mode and 5 min in deep sleep mode. Consequently, the node is unable to monitor or read the sensor while in sleep mode. Fire occurring during sleep mode will not be detected; it can only be detected during running mode. This explains the difference in response time of the IoT during the testing. Fig. 15 illustrates the position of the IoT sensing node at the time of fire ignition. The IoT response time is the duration from when the fire turns up to when the IoT node sends an alarm response to the LoRaWAN Gateway, which then forwards it to the cloud server. The test is repeated 10 times. Furthermore, the drone's flight time to reach the fire spot is also determined, as shown in the table below. The drone is located 5 km away from the IoT sensing node, and all tests were conducted on the same day when the wind speed was 12 km/h.

Fig. 15. IoT sensing node location while fire is ignited.

Table 14
Response times of IoT fire sensing nodes and drones.

Experiment #	IoT fire sensing node response time (min)	Time taken for the drone to reach the fire (min)
1	3:35	4:45
2	2:15	4:56
3	4:13	5:00
4	1:45	4:41
5	2:50	4:49
6	1:19	4:47
7	4:30	5:05
8	3:20	4:43
9	3:45	4:49
10	2:44	4:57

Table 15
Confusion matrix.

		Predicted label	
		Fire	No Fire
True label	Fire	99.2%	0.8%
	No Fire	0.3%	99.7%

4.5. Object classification and fire tracking results

To comprehensively evaluate the performance of our proposed drone-based fire detection classifier, we utilized several key metrics: accuracy, precision, recall, and the F1 measure. To quantitatively validate the efficacy of our visual fire classification algorithm, we trained the classifier on a dataset comprising 5,054 images, which were categorized into two distinct classes: "Fire" and "NoFire". The precision–recall curve, depicted in Fig. 16, provides a visual representation of the trade-off between precision and recall for our trained network. Additionally, the confusion matrix presented in Table 15 offers detailed insights into the classifier's performance across different categories, further enabling the assessment of its accuracy, precision, recall, and F1 score. This multifaceted evaluation approach ensures a rigorous validation of our model's classification capabilities.

We compute the performance metrics and tabulate them in Table 16. The proposed classifier achieved a classification accuracy of 99.46% and a mean average precision of 99.64%. Table 17 shows the superiority of our proposed CNN model compared to state-of-the-art methods in terms of accuracy tested on the FLAME dataset. Shamsoshoara et al. introduced the FLAME dataset, supporting machine learning algorithm development for fire detection and segmentation, and presented an ANN and a U-Net-based method, achieving high precision (92%) and recall (84%) [29]. Dutta and Ghosh proposed an innovative architecture that combines separable CN) with digital image processing techniques such as thresholding and segmentation. Similarly, Treneska and Stojkoska explored the application of transfer learning to improve wildfire detection accuracy in UAV-collected images from the FLAME dataset. They fine-tuned various CNN architectures, with the ResNet50 model achieving the highest accuracy of 88%, demonstrating the potential of transfer learning for enhancing model generalization in real-time wildfire detection applications [34].

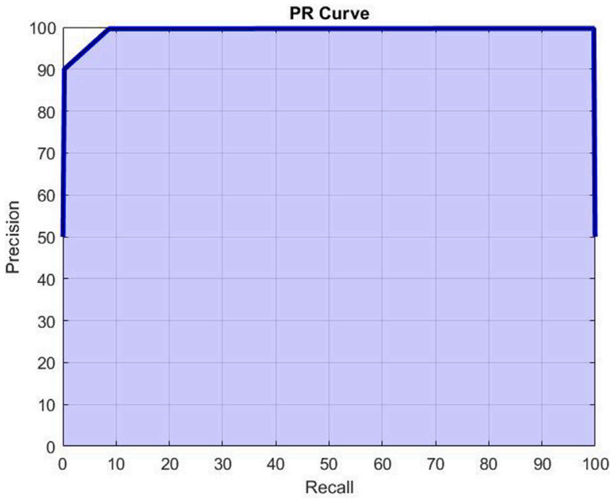

Fig. 16. Precision vs Recall curve for our proposed classifier.

Table 16
Fire classification performance metrics.

Performance measure	Score
Accuracy	99.46%
Precision	99.64%
Recall	99.29%
F1-Score	99.46%

Table 17
Comparison with state-of-the-art CNN-based approaches for fire detection on the FLAME dataset.

Approach	Accuracy (%)
Shamsoshoara et al. [29]	76.23
Ghali et al. [35]	85.12
Treneska et al. [34]	88.01
Zhang et al. [36]	79.48
Our Model	99.46

Further contributions include the work of Ghali et al. who developed an ensemble learning method combining EfficientNet-B5 and DenseNet-201 for classification and employed vision transformers like TransUNet and TransFire for segmentation, achieving superior accuracy and efficiency [35]. Zhang et al. also utilized transfer learning with a fine-tuned ResNet50 model, achieving a recognition accuracy of 79.48%, addressing challenges such as limited UAV-labeled images and varying visual conditions [36].

The model introduced in [34] leverages a pre-trained ResNet50 as a feature extractor, incorporating global average pooling and dropout for a streamlined yet powerful classification framework. The authors in [29] employ the Xception network, known for its depthwise separable convolutions and residual connections, providing a robust feature extraction mechanism. The model in [35] combines DenseNet201 and EfficientNetB5, using dual networks to capture a wide range of features with data augmentation to improve robustness. The proposed model in [36] integrates traditional and mixup-based augmentation techniques with a multi-stage architecture, pre-trained on ImageNet, to achieve comprehensive feature extraction and classification, utilizing focal loss to address class imbalance.

In comparison, our proposed CNN-based fire classifier adopts a meticulous approach by utilizing a series of Conv2D and DepthwiseConv2D layers organized into structured blocks, progressively downsampling the input while expanding feature maps. The architecture emphasizes efficiency through depthwise separable convolutions and integrates batch normalization and ReLU activations to stabilize training. Additionally, our model includes global average pooling and a fully connected layer with softmax activation for final classification. This method balances complexity and computational efficiency, aiming to achieve precise fire detection with a detailed and structured feature extraction process, making it a robust and effective alternative to the aforementioned models.

Fig. 17 shows a demonstration of the proposed fire tracking algorithm where the particles reflect the location of the fire. As the fire expands, i.e. the location of the particles change within the frame, the drone receives the updated coordinates and flies around, ensuring the center of the fire is within the camera's i.e., changes field of view.

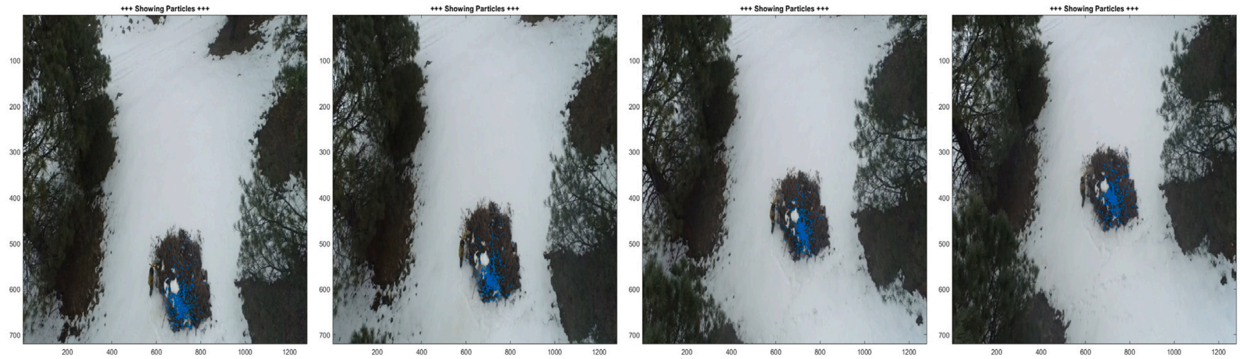

Fig. 17. Demonstration of the fire tracking algorithm where the particles are initially detected within the FOV but far from the center of the frame, requiring the drone to update its location to ensure the center of the fire is tracked.

5. Limitations and future work

While technologically advanced, the proposed UIoT system encounters several deployment challenges. The approach introduced in this work must adhere to diverse environmental and telecommunications regulations. The proposed system relies on IoT nodes to detect the presence of fire and AI-powered drones for fire localization and management. This number of IoT nodes determines the system's scalability, although environmental integration might pose challenges. The work introduces a cost-effective solution. However, the maintenance cost of such technology, which is exposed to many environmental factors and adverse weather conditions, would be a logistical challenge. The interference of the LoRaWAN signals with other wireless devices in the same bandwidth could degrade communication reliability. Our system's resilience to the mentioned environmental integration challenges is yet to be validated.

Integrating our system with the regulatory bodies and existing wildfire management frameworks will establish compliance standards and protocols that facilitate the system's adoption within current legal frameworks. Additionally, efforts to enhance cost-effectiveness are vital, encompassing optimizing system design to minimize expenses and exploring robust maintenance protocols, particularly for IoT nodes deployed in remote areas. This includes the development of low-cost, low-maintenance fire-detecting IoT nodes and strategies to streamline their upkeep.

Investigations into adaptive technologies capable of withstanding adverse weather conditions and challenging landscapes are essential. These efforts should include research into materials and designs that enhance the durability and reliability of the UIoT components. Frequency management and advanced encoding techniques are necessary to minimize interference with other wireless devices operating near the LoRaWAN channels.

6. Conclusions

This paper proposes a UAV-IoT system design for wildfire detection. Our system is based on low-cost and low-maintenance fire-detecting IoT nodes that communicate using LoRaWAN and HTTPS communication protocols with our cloud server to report the sensed measurements. Once a fire is detected at a specific location, an autonomous drone with a visual camera flies toward the identified geolocation of the node to monitor the fire's progress. The drone runs our novel fire center detection and tracking algorithm for precise localization and then uses an appropriate countermeasure to extinguish the fire. The proposed system uses static gateways connecting neighboring sensing nodes architecture for system integration. Authorities receive detailed real-time reports on the fire's location, progress, and the countermeasures taken. Numerical results show that the UAV-IoT system can be a cost-efficient alternative to satellite imaging for wildfire detection, especially when the cost of fire-relevant losses is high.

CRediT authorship contribution statement

Montaser N.A. Ramadan: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation. **Tasnim Basmaji:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation. **Abdalla Gad:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation. **Hasan Hamdan:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation. **Bekir Tevfik Akgün:** Writing – review & editing, Supervision. **Mohammed A.H. Ali:** Writing – review & editing, Supervision. **Mohammad Alkhedher:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Mohammed Ghazal:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

The authors acknowledge financial support from Abu Dhabi University's Office of Research and Sponsored Programs. Grant number: 19300792.

Appendix. Particle filter algorithm

The algorithm model has the form

$$S_n = f(S_{n-1}) + U_n, \tag{A.1}$$

where S_n represents the state at time n. f() is the transition function of the algorithm model, and U_n is the algorithm noise. This model is used to predict the possible position hypotheses of the object of interest, i.e. fire.

The observation model is used to estimate the likelihood of each position hypothesis and determine the one with the highest likelihood. It is given by:

$$Y_n = M(S_n) + V_n, \tag{A.2}$$

where M() represents the measurement function and V_n is the noise. Let S_n and Y_n be the hidden state of the object of interest and measurements at the discrete time n, respectively. For object tracking, the distribution of interest is the $p(S_n|Y_{1:n})$, also known as filtering distribution, where $Y_{1:n} = (Y_1, \ldots, Y_n)$ denotes all of the observations up to the current time step n. In Bayesian sequential estimation, the filtering distribution can be computed according to the two step recursion: prediction

$$p(S_n|Y_{1:n-1}) = \int p(S_n|S_{1:n-1})p(S_{n-1}|Y_{1:n-1})dS_{n-1}, \tag{A.3}$$

and filtering

$$p(S_n|Y_{1:n}) = p(Y_n|S_{1:n})p(S_n|Y_{1:n-1}) \tag{A.4}$$

This recursion requires the specification of a dynamic state transition model describing the state evolution $p(S_n|Y_{1:n-1})$ and a model that gives the likelihood of any state in the light of the current observation, $p(Y_n|S_{1:n})$. The recursion is initialized with some distribution for the initial state $p(S_0)$. Once the sequence of filtering distribution is known, point estimates of the state can be obtained according to any appropriate function, leading, for example, to the Maximum a Posteriori (MAP) estimate, $arg\,max_{S_n} p(S_n|Y_{1:n-1})$, and the Minimum Mean Square Error (MMSE) estimate. For particle filter tracking, we begin with a weighted set of samples $\{S_{n-1}^{(i)}, W_{k-1}^{(i)}\}_{i=1}^{N}$ approximately distributed according to $p(S_{n-1}|Y_{1:n-1})$, new samples are generated from a suitable proposal distribution, which may depend on the previous state and the new measurements, i.e., $S_n^{(i)} : q_p(S_n|S_{n-1}^{(i)}, Y_n)$, i = 1, …, N. To maintain a consistent sample, the new importance weights are set

$$W_k^{(i)} = W_{k-1}^{(i)} \frac{p(Y_n|S_n^{(i)})p(S_n^{(i)}|S_{n-1}^{(i)})}{q_p(S_n|S_{n-1}^{(i)}, Y_n)}, \tag{A.5}$$

where $W_{k-1}^{(i)}$ represents the weight at the previous time step for particle i and $\sum_{i=1}^{N} W_k^{(i)} = 1$. The new particle set $\{S_n^{(i)}, W_n^{(i)}\}_{i=1}^{N}$ is then approximately distributed according to $p(S_n|Y_{1:n})$. The observation model reflects which image features will be used to update the current state estimate. We use the color distributions as the target observation models to achieve robustness against non-rigidity, rotation, and partial occlusion. The tracking process is initiated in case of a positive class detection. We convert the fire's detected position in pixels to real-world coordinates in meters for accurate fire localization as detailed in Eqs. (A.10)–(A.13). Let u and v represent the image coordinates in pixels, x and y denote the coordinates of the image coordinate system in millimeters, f represents the focal length, and X and Y denote the coordinates in meters. The relationship between the pixel coordinate system and the image plane coordinate system is known to be as follows:

$$u = \frac{x}{dx} + u_0 \tag{A.6}$$

$$v = \frac{y}{dy} + v_0 \tag{A.7}$$

which can be expressed as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & 1 & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{A.8}$$

The relationship between the camera coordinate system and the real-world coordinate system can be explained as

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = C \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{A.9}$$

The following relationship can then be obtained between the focal length, image and camera coordinate systems.

$$\frac{x}{f} = \frac{X_c}{Z_c} \tag{A.10}$$

$$\frac{y}{f} = \frac{Y_c}{Z_c} \tag{A.11}$$

which can be expressed as follows:

$$Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{A.12}$$

Finally, we compute the relationship between the real-world coordinate system and the pixel coordinate system as follows:

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & s & u_0 & 0 \\ 0 & \frac{1}{dy} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = C_1 C_2 X = CX \tag{A.13}$$

where C is the projection matrix, C_1 and C_2 are the camera internal ($\frac{f}{dx}$, $\frac{f}{dy}$, u_0, v_0, and s) and external parameters (R and T) [37]. These coordinates are then fed into a coordinate transform to generate the object's predicted GPS as explained in Eqs. (A.14)–(A.20).

Given that X and Y represent the fire's current location calculated from the perspective of the image in meters. Let θ_d represent the bearing referenced to the True North in Radians, d_d represents the distance between the fire and the drone's location, R represents the Earth's radius (6378.1), and $Drone_{lat}$, $Drone_{lon}$, and $Drone_{alt}$ indicate the drone's latitude, longitude, and altitude respectively. Hence, $Fire_{lat}$, $Fire_{lon}$, and $Fire_{alt}$ provide the center of fire's current location as GPS coordinates.

$$\theta_R = \theta_d * (\frac{\pi}{180}) \tag{A.14}$$

$$Drone_{latR} = Drone_{lat} * (\frac{\pi}{180}) \tag{A.15}$$

$$Drone_{lonR} = Drone_{lon} * (\frac{\pi}{180}) \tag{A.16}$$

$$Fire_{latR} = asin(sin(Drone_{latR}) * cos(\frac{d_d}{R}) + cos(Drone_{latR}) * sin(\frac{d_d}{R}) * cos(\theta_R)) \tag{A.17}$$

$$Fire_{lonR} = Drone_{lon} + atan(\frac{sin(\theta_R) * sin(\frac{d}{R}) * cos(Drone_{latR})}{cos(\frac{d_d}{R}) - sin(Drone_{latR}) * sin(Fire_{latR})}) \tag{A.18}$$

$$Fire_{lat} = Fire_{latR} * \frac{180}{\pi} \tag{A.19}$$

$$Fire_{lon} = Fire_{lonR} * \frac{180}{\pi} \tag{A.20}$$

References

[1] K. Hoover, L.A. Hanson, IF10244, Technical Report, 2022, Congressional Research Service.

[2] E. Commission, J.R. Centre, J. San-Miguel-Ayanz, T. Durrant, R. Boca, P. Maianti, G. Libertà, T. Artés Vivancos, D. Oom, A. Branco, et al., Forest Fires in Europe, Middle East and North Africa 2020, Publications Office of the European Union, 2021, http://dx.doi.org/10.2760/216446.

[3] S. Li, T. Banerjee, Spatial and temporal pattern of wildfires in California from 2000 to 2019, Sci. Rep. (2021) 11, http://dx.doi.org/10.1038/s41598-021-88131-9.

[4] D. Wang, D. Guan, S. Zhu, M.M. Kinnon, G. Geng, Q. Zhang, H. Zheng, T. Lei, S. Shao, P. Gong, et al., Economic footprint of California wildfires in 2018, Nature Sustainability 4 (2021) 252–260, http://dx.doi.org/10.1038/s41893-020-00646-7.

[5] K. Nakau, M. Fukuda, K. Kushida, H. Hayasaka, K. Kimura, H. Tani, Forest Fire Detection Based on MODIS Satellite Imager Y, and Comparison of NOA a Satellite Imagery with Fire Fighters' Information, Technical Report.
[6] E.I. Ponomarev, V. Ivanov, N. Korshunov, Chapter 10 - System of Wildfires Monitoring in Russia, Elsevier, 2015, pp. 187–205, http://dx.doi.org/10.1016/B978-0-12-410434-1.00010-5.
[7] F.I. Brown, W. Schroeder, A. Setzer, M.M. de Los Rios, N. Pantoja, A. Duarte, J. Marengo, Monitoring fires in southwestern Amazonia rain forests, Eos 87 (2006) 253–259, http://dx.doi.org/10.1029/2006EO260001.
[8] N. Maeda, H. Tonooka, Early stage forest fire detection from Himawari-8 AHI images using a modified MOD14 algorithm combined with machine learning, Sensors 23 (2023) http://dx.doi.org/10.3390/s23010210, Article 210.
[9] E. Jang, Y. Kang, J. Im, D.-W. Lee, J. Yoon, S.-K. Kim, Detection and monitoring of forest fires using Himawari-8 geostationary satellite data in South Korea, Remote Sens. 11 (2019) http://dx.doi.org/10.3390/rs11030271, Article 271.
[10] Y. Zheng, G. Zhang, S. Tan, Z. Yang, D. Wen, H. Xiao, A forest fire smoke detection model combining convolutional neural network and vision transformer, Front. Forests Global Change (2023) 6, http://dx.doi.org/10.3389/ffgc.2023.1136969.
[11] H. Xu, G. Zhang, R. Chu, J. Zhang, Z. Yang, X. Wu, H. Xiao, Detecting forest fire omission error based on data fusion at subpixel scale, Int. J. Appl. Earth Obs. Geoinf. 128 (2024) http://dx.doi.org/10.1016/j.jag.2024.103737, Article 103737.
[12] M. Rodrigues, F. Alcasena, C. Vega-García, Modeling initial attack success of wildfire suppression in Catalonia, Spain, Sci. Total Environ. 666 (2019) 915–927, http://dx.doi.org/10.1016/j.scitotenv.2019.02.323.
[13] R. Hansen, Aerial suppression penetrating an axially symmetric and upright buoyant wildfire plume, Int. J. Saf. Secur. Eng. 9 (2019) 287–304, http://dx.doi.org/10.2495/SAFE-V9-N4-287-304.
[14] M.P. Plucinski, A.L. Sullivan, R.J. Hurley, A methodology for comparing the relative effectiveness of suppressant enhancers designed for the direct attack of wildfires, Fire Saf. J. 87 (2017) 71–79, http://dx.doi.org/10.1016/j.firesaf.2053312.005.
[15] T. Zhou, J. Lu, L. He, C. Wu, J. Luo, Experiments in aerial firefighting with and without additives and its application to suppress wildfires near electrical transmission lines, J. Fire Sci. 07349041221098171, http://dx.doi.org/10.1177/07349041221098171.
[16] R. Mahaveerakannan, C. Anitha, A.K. Thomas, S. Rajan, T. Muthukumar, G.G. Rajulu, An IoT based forest fire detection system using integration of cat swarm with LSTM model, Comput. Commun. 211 (2023) 37–45, http://dx.doi.org/10.1016/j.comcom.2023.08.020.
[17] A.A. Siddique, N. Alasbali, M. Driss, W. Boulila, M.S. Alshehri, J. Ahmad, Sustainable collaboration: Federated learning for environmentally conscious forest fire classification in green internet of things (IoT), Internet Things 25 (2024) http://dx.doi.org/10.1016/j.iot.2023.101013, Article 101013.
[18] A. Sharma, P.K. Singh, Y. Kumar, An integrated fire detection system using IoT and image processing technique for smart cities, Sustainable Cities Soc. 61 (2020) http://dx.doi.org/10.1016/j.scs.2020.102332, Article 102332.
[19] A. Lertsinsrubtavee, T. Kanabkaew, S. Raksakietisak, Detection of forest fires and pollutant plume dispersion using IoT air quality sensors, Environ. Pollut. 338 (2023) http://dx.doi.org/10.1016/j.envpol.2023.122701, Article 122701.
[20] F. De Vivo, M. Battipede, E. Johnson, Infra-red line camera data-driven edge detector in UAV forest fire monitoring, Aerosp. Sci. Technol. 111 (2021) http://dx.doi.org/10.1016/j.ast.2021.106574, Article 106574.
[21] Y. Qiao, W. Jiang, F. Wang, G. Su, X. Li, J. Jiang, FireFormer: An efficient transformer to identify forest fire from surveillance cameras, Int. J. Wildland Fire (2023) 32, http://dx.doi.org/10.1071/WF22220.
[22] M. Jemmali, B.M. Loai Kayed, W. Boulila, H. Amdouni, M.T. Alharbi, Optimizing forest fire prevention: Intelligent scheduling algorithms for drone-based surveillance system, Procedia Comput. Sci. 225 (2023) 1562–1571, http://dx.doi.org/10.1016/j.procs.2023.10.145, 27th International Conference on Knowledge Based and Intelligent Information and Engineering Systems (KES 2023).
[23] T. Rathod, V. Patil, R. Harikrishnan, P. Shahane, Multipurpose deep learning-powered UAV for forest fire prevention and emergency response, HardwareX 16 (2023) http://dx.doi.org/10.1016/j.ohx.2023.e00479, Article e00479.
[24] S. Sudhakar, V. Vijayakumar, C. Sathiya Kumar, V. Priya, L. Ravi, V. Subramaniyaswamy, Unmanned aerial vehicle (UAV) based forest fire detection and monitoring for reducing false alarms in forest-fires, Comput. Commun. 149 (2020) 1–16, http://dx.doi.org/10.1016/j.comcom.2019.10.007.
[25] T. Marques, S. Carreira, R. Miragaia, J. Ramos, A. Pereira, Applying deep learning to real-time UAV-based forest monitoring: Leveraging multi-sensor imagery for improved results, Expert Syst. Appl. 245 (2024) http://dx.doi.org/10.1016/j.eswa.2023.123107, Article 123107.
[26] Z. Xue, H. Lin, F. Wang, A small target forest fire detection model based on YOLOv5 improvement, Forests 13 (2022) http://dx.doi.org/10.3390/f13081332, Article 1332.
[27] G. Georgiades, X.S. Papageorgiou, S. Loizou, Integrated forest monitoring system for early fire detection and assessment, in: Proceedings of the 2019 6th International Conference on Control, Decision and Information Technologies, CoDIT, Paris, France, 23-26, 2019, pp. 1817–1822, http://dx.doi.org/10.1109/CoDIT.2019.8820548.
[28] N. Ya'acob, M.S.M. Najib, N. Tajudin, A.L. Yusof, M. Kassim, Image processing based forest fire detection using infrared camera, J. Phys. Conf. Ser. 1768 (2021) http://dx.doi.org/10.1088/1742-6596/1768/1/012014, Article 012014.
[29] A. Shamsoshoara, F. Afghah, A. Razi, L. Zheng, P.Z. Fulé, E. Blasch, Aerial imagery pile burn detection using deep learning: The FLAME dataset, Comput. Netw. (ISSN: 1389-1286) 193 (1080) (2021) 01, http://dx.doi.org/10.1016/j.comnet.2021.108001.
[30] B. Zoph, Q.V. Le, Neural architecture search with reinforcement learning, 2017, arXiv arXiv:1611.01578.
[31] S. Konatowski, P. Kaniewski, J. Matuszewski, Comparison of estimation accuracy of EKF, UKF and PF filters, Ann. Navig. 23 (2016) 69–87.
[32] I. Ullah, X. Su, X. Zhang, D. Choi, Z. Hou, J. Zhu, Evaluation of localization by extended Kalman filter, unscented Kalman filter, and particle filter-based techniques, Wirel. Commun. Mob. Comput. 2020 (2020) http://dx.doi.org/10.1155/2020/8898672, Article 8898672.
[33] S. Arulampalam, B. Ristić, Comparison of the particle filter with range-parameterized and modified polar EKFs for angle-only tracking, Proc. SPIE 4048 (2000) 288–299, http://dx.doi.org/10.1117/12.391985.
[34] S. Treneska, B.R. Stojkoska, Wildfire detection from UAV collected images using transfer learning, in: Proceedings of the 18th International Conference on Informatics and Information Technologies, Xi'an, China, 12–14, 2021, pp. 6–7.
[35] R. Ghali, M.A. Akhloufi, W.S. Mseddi, Deep learning and transformer approaches for UAV-based wildfire detection and segmentation, Sensors 22 (2022) 1977, http://dx.doi.org/10.3390/s22051977.
[36] L. Zhang, M. Wang, Y. Fu, Y. Ding, A forest fire recognition method using UAV images based on transfer learning, Forests 13 (2022) 975, http://dx.doi.org/10.3390/f13070975.
[37] W. Xue-Jun, G. Dong-Yuan, Y. Xi-Fan, Application of matlab calibration toolbox for camera's intrinsic and extrinsic parameters solving, in: Proceedings of the 2019 International Conference on Smart Grid and Electrical Automation, ICSGEA, 2019, pp. 106–111, http://dx.doi.org/10.1109/ICSGEA.2019.00032.