



Article

A Cyber-Physical System for Wildfire Detection and Firefighting

Pietro Battistoni ¹, Andrea Antonio Cantone ¹, Gerardo Martino ¹, Valerio Passamano ¹,
Marco Romano ^{2,*}, Monica Sebillio ¹ and Giuliana Vitiello ¹

- ¹ Department of Computer Science, University of Salerno, 84084 Fisciano, Italy; pbattistoni@unisa.it (P.B.); acantone@unisa.it (A.A.C.); g.martino11@studenti.unisa.it (G.M.); v.passamano1@studenti.unisa.it (V.P.); msebillio@unisa.it (M.S.); gvitiello@unisa.it (G.V.)
- ² Faculty of Political Science and Psychosocial Studies, Università degli Studi Internazionali di Roma—UNINT, 00147 Roma, Italy
- * Correspondence: marco.romano@unint.eu

Abstract: The increasing frequency and severity of forest fires necessitate early detection and rapid response to mitigate their impact. This project aims to design a cyber-physical system for early detection and rapid response to forest fires using advanced technologies. The system incorporates Internet of Things sensors and autonomous unmanned aerial and ground vehicles controlled by the robot operating system. An IoT-based wildfire detection node continuously monitors environmental conditions, enabling early fire detection. Upon fire detection, a UAV autonomously surveys the area to precisely locate the fire and can deploy an extinguishing payload or provide data for decision-making. The UAV communicates the fire's precise location to a collaborative UGV, which autonomously reaches the designated area to support ground-based firefighters. The CPS includes a ground control station with web-based dashboards for real-time monitoring of system parameters and telemetry data from UAVs and UGVs. The article demonstrates the real-time fire detection capabilities of the proposed system using simulated forest fire scenarios. The objective is to provide a practical approach using open-source technologies for early detection and extinguishing of forest fires, with potential applications in various industries, surveillance, and precision agriculture.



Citation: Battistoni, P.; Cantone, A.A.; Martino, G.; Passamano, V.; Romano, M.; Sebillio, M.; Vitiello, G. A Cyber-Physical System for Wildfire Detection and Firefighting. *Future Internet* **2023**, *15*, 237. <https://doi.org/10.3390/fi15070237>

Academic Editors: Guan Gui, Yun Lin and Haitao Zhao

Received: 30 May 2023
Revised: 2 July 2023
Accepted: 3 July 2023
Published: 6 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: CPS; IoT; IoRT; UAV; UGV; ROS; wildfire; fire detection; Node-RED; MQTT

1. Introduction

Forests are crucial ecosystems that provide numerous benefits to the environment and its inhabitants. They filter air and water, provide food and shelter for animals, and play a crucial role in regulating the climate. However, global warming and human error have increased the incidence of forest wildfires, which can cause significant damage to the environment, property, and human health. Once an ignition starts, it is essential to declare it as soon as possible so that it can be rapidly controlled and suppressed. Researchers have focused on providing first responders, firefighters, and decision makers with systems for analyzing the situation at a glance through visual analytics techniques [1–4], sharing data [1], or systems that involve active citizenship for early warning [5,6]. However, wild-land firefighters often do not have access to accurate and real-time information about the situation because the location of the fire is remote or hard to reach, and if the fire is not brought under control in time, it can continue to grow, increasing the danger and the suppression costs. Detecting and controlling wildfires is a challenging task, as they can spread rapidly and are difficult to anticipate. Researchers are exploring the use of advanced technologies such as IoT, robotics, and drone technologies [7]. The synergy of these technologies converges in a new trend that can be called the Internet of Robotics and Drone Things [8]. The interconnection of all these heterogeneous devices and the integration with software components has led to the design and development of a Cyber-Physical System (CPS) for

wildfire detection [9] and collaborative robots able to navigate autonomously and perform fire attacks on wildland fires before they burn out of control. However, there remains a significant challenge in providing accurate and real-time information to ground control stations and operators overseeing fleets of drones in collaborative environments. In [10], the authors introduce a human–machine interface that enables a ground control station to remotely monitor and manage a fleet of drones in a collaborative environment, with the involvement of multiple operators. Collaborative robots, also defined as unmanned systems, have the capability to execute a mission autonomously without human intervention, and commonly they include unmanned aircraft and ground robots [11].

Proposed CPS for Wildfire Detection

Building upon this research, our project aims to design, simulate, and implement a Cyber-Physical System (CPS) for wildfire detection using low-cost Internet of Things (IoT) sensors and Robot Operating System (ROS)-based collaborative unmanned robots for fire extinguishing. It is a cutting-edge solution designed to address the increasingly urgent issue of wildfires, which are becoming more frequent and severe due to the effects of climate change. By utilizing low-cost IoT sensors to continuously monitor temperature and humidity, the system is able to detect potential fire hazards at an early stage, such as dry vegetation or other combustible materials, and take appropriate measures to prevent a serious threat. The system also incorporates a smart camera and infrared sensor, enabling real-time detection of flames, which is crucial for quickly identifying and responding to wildfires [12]. Additionally, a Carbon Dioxide (CO₂) sensor provides real-time monitoring of carbon monoxide in the air, which can help measure the severity of the fire and the level of danger it poses to nearby communities. The data collected by the IoT system is sent to a ground forest fire monitoring and alert station, which is able to visualize the data through a web-based dashboard, allowing for efficient and effective decision-making in response to a wildfire. This system can also trigger appropriate response measures, such as sending alerts to local authorities and emergency services, as well as unmanned systems, for an immediate and coordinated response. The integration of unmanned systems, also known as drones or Unmanned Aerial Vehicles (UAVs), will give the advantage of quickly identifying the exact fire spots after a navigation survey over the wildland and then releasing a ball extinguisher to suppress the fires [13]. In addition, the UAV will transmit the fire spot coordinates to an Unmanned Ground Vehicle (UGV), also called a robot rover, that will empower the overall architecture, by automatically navigating into the forest, reaching the fire spots, and suppressing the fire using water or chemical extinguisher [14].

Our work contributes to the scientific knowledge base by introducing a CPS architecture specifically designed for wildfire detection and firefighting. This architecture is based on the concept of the Internet of Robotic Things (IoRT) [15], integrating a combination of cutting-edge technologies, including IoT sensors, UAVs, and UGVs, while utilizing a set of open-source solutions that are widely popular and accessible to all. This integration enables the development of a scalable wildfire prevention system.

The IoRT network, as a specific instantiation of the Internet of Things (IoT) in the context of robotic systems, facilitates seamless collaboration and information exchange between robotic devices and other IoT components. In the context of forestry tasks, this concept holds particular relevance as it allows us to utilize robotic devices to respond to fires detected by the IoT network. One significant advantage of this architecture is its ability to overcome the limitations of existing firefighting systems by integrating and orchestrating heterogeneous robotic systems and user interfaces for data monitoring. In contrast to previous systems that focused on isolated, manually operated approaches [16], our research focuses on the development of a collaborative ecosystem [14] capable of autonomously managing forest fires. To illustrate the practical implementation of this architecture, we have developed a prototype that underwent simulation experiments.

While the system development process and simulation procedures are thoroughly described in this study, and the simulation results demonstrate the effectiveness of the

proposed architecture, it is important to acknowledge that our CPS system is currently in the developmental phase. As a result, there are certain limitations in evaluating its performance in a real-world environment. Factors such as communication capabilities, terrain accessibility, and atmospheric conditions in which the robots can effectively operate are crucial aspects that require further investigation and evaluation in the real world.

By presenting this CPS architecture and its simulation outcomes, our goal is to establish a solid foundation for future research and development in the field of wildfire prevention and management based on IoRT. This approach can enable fire stations and other wildfire suppression organizations to adopt a more efficient and effective approach to combating wildfires through the combination of innovative technologies and a collaborative framework.

The paper is structured as follows: Section 2 examines some related work. Section 3 presents the design and implementation of the cyber-physical system for wildfire detection. Section 4 presents a simulation, and Section 5 presents a discussion. Finally, Section 6 draws conclusions, and Section 7 presents future research.

2. Related Work

In recent years, with global temperature rise, extreme weather, and climate events, especially droughts, wildfires are a natural phenomenon that has increased in frequency and magnitude over the past decade, causing damage to life and property, and impacting climate and air quality [17]. The wildfire season has lengthened in many countries due to shifting climate patterns. Studies report that the number of wildfires has declined, thanks to the strict and full implementation of forest fire prevention and management strategies and the philosophy of forest protection and green development in the world; however, climate warming will continue in the world, leading to the more frequent occurrence of dry thunderstorms, increased number of days with forest fire alerts, higher fire risks, and greater challenges in forest fire prevention [18]. Therefore, we must strengthen studies on the impacts of climate change on forest fires; improve the monitoring, prediction, and alert system for forest fires; and raise public awareness and knowledge of forest fire prevention [19]. To minimize their impact, effective prevention, early warning, and response approaches are needed. Research is ongoing to develop high-accuracy fire detection systems for challenging environments [20]. Traditionally, forest fires were mainly detected by human observation from fire lookout towers and involved only primitive tools, such as the Osborne fire Finder [21], which is a tool consisting of a card topographic printed on a disc with edge graduated. Unfortunately, these primary techniques are inefficient due to the unreliability of human observation towers and difficult living conditions. Advanced technologies such as satellites, UAV, UGV, ground-based sensor nodes, and camera systems supplement traditional firefighting techniques, focusing on detecting wildfires at early stages, and predicting hot spots by combining all the methods for a robust fire monitoring system [22]. Cyber-Physical Systems (CPSs), which integrate computing and control technologies into physical systems aim to contribute to real-time wildfire detection through the Internet of Things (IoT) wireless sensor [23] and the use of a collaborative unmanned system [11].

In the past few years, IoT has been gaining significant attention in various fields, including forest fire detection. IoT technologies are very useful in forest management systems because they can provide real-time monitoring of forest conditions, such as temperature, humidity, and wind speed. These data can be used to predict the likelihood of a wildfire, to detect a fire, and to track the progression of an active fire. This kind of technology is also called the Internet of Forest Things (IoFT) [24]. The Milesight Company is an example of a company that has developed a technology to detect and prevent forest fires using an IoT system (Figure 1). The system is comprised of:

- **IoT nodes:** A microcontroller unit connected with carbon dioxide, barometric pressure, temperature, and humidity sensors.

- **Gateway:** A central hub that connects and manages the various sensors and devices, is able to collect data from IoT nodes, transmit it to the cloud, and perform edge computing tasks.
- **Cloud:** A remote server that provides a scalable and flexible platform used to process, and analyze the data collected by the sensors, but also used to store data, provides visualization and reporting tools and enables integration with other systems for enhanced capabilities.

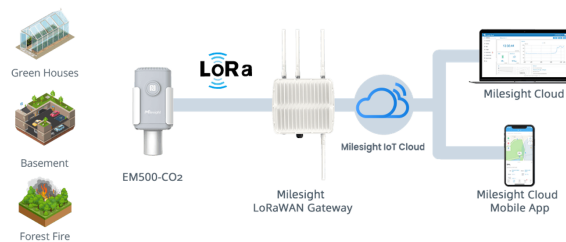


Figure 1. Architecture of forest fire detection.

Typically, the IoT nodes used in this type of application employ Low-Power Wide-Area Network (LPWAN) technologies, such as LoRa and Sigfox, to enable long-range communication with a battery life that can last for several years. This allows for coverage of an entire forest at a cost-effective price. Additionally, the use of unmanned systems offers many advantages, including the ability to cover large areas in any weather condition, operate during both day and night with extended flight duration, be quickly recovered and relatively cost-effective compared to other methods, carry various payloads for different missions within a single flight, and efficiently cover larger and specific target areas [25]. UAVs equipped with computer vision-based remote sensing systems are becoming a more practical option for forest fire surveillance and detection due to their low cost, safety, mobility, and rapid response characteristics [26]. Unmanned Aerial Vehicles (UAVs) based on computer vision fire detection can provide immediate information on the fire spots but are also equipped with a dropping mechanism used to target and shoot the fire by releasing extinguishing balls [13,27]. Popular real-time object detection systems, such as YOLO can be integrated into UAVs, providing real-time monitoring of large forest areas and detecting fires at an early stage, reducing the risk of the fire spreading [28]. Consequently, numerous research studies have been conducted in recent years to develop UAV-based forest fire monitoring and detection applications based on YOLO [12,29,30]. In addition, autonomous ground systems such as the UGV are essential in responding to wildfire emergencies. Equipped with advanced technologies such as computer vision, infrared and depth cameras, and other sensors, UGVs can be used for detecting and extinguishing fires [31–33], as well as cooperating with other systems such as UAVs [14]. These robots can perform tasks on land without human intervention and access difficult-to-reach areas that may be too hazardous for human firefighters.

Forest fire monitoring and detection applications using Unmanned Aerial Vehicles (UAVs) have received significant attention in recent years, with a particular focus on employing the You Only Look Once (YOLO) algorithm for object detection [12,29,30]. These studies have demonstrated the effectiveness of UAVs in providing aerial surveillance and timely fire detection capabilities. Furthermore, autonomous ground systems such as Unmanned Ground Vehicles (UGVs) have emerged as crucial elements in responding to wildfire emergencies, leveraging advanced technologies such as computer vision, infrared and depth cameras, and other sensors [31–33]. UGVs are capable of detecting and extinguishing fires and collaborating with UAVs to access hazardous or hard-to-reach areas that pose risks to human firefighters [14]. Object detection algorithms play a critical role in fire detection systems as they enable real-time identification and localization of fire incidents. The You Only Look Once (YOLO) family of object detection algorithms has proven highly effective and efficient in computer vision applications. Notably, YOLOv4 (<https://github.com/yolov4>):

[//blog.roboflow.com/a-thorough-breakdown-of-yolov4/](https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/), accessed on 19 April 2023) and YOLOv5 (<https://blog.roboflow.com/yolov5-improvements-and-evaluation/>, accessed on 19 April 2023) stand out as prominent versions, each offering distinct advantages for different components as shown in Table 1. In our study, we integrated YOLOv4 into the UGV component, leveraging its state-of-the-art accuracy and compatibility with the darknet framework and Robot Operating System (ROS). This integration facilitated object detection and localization within the UGV, enabling seamless collaboration within the overall system architecture. To extend the capabilities of our system, we incorporated YOLOv5 into the UAV component. YOLOv5 features a streamlined architecture and optimized data augmentation strategies, making it ideal for resource-constrained UAVs. This integration empowered the UAV to perform real-time fire detection and localization, enabling efficient aerial surveillance and coordination with the UGV. By harnessing the strengths of YOLOv4 and YOLOv5 in our collaborative UGV and UAV system, we achieved robust object detection capabilities, seamless integration within the system framework, and real-time fire detection and localization. Our study builds upon the advancements made by previous research studies utilizing YOLO in UAV-based fire detection applications while also showcasing the effective integration of UGVs in firefighting operations.

Table 1. Comparison of YOLOv4 and YOLOv5.

| Feature | YOLOv4 | YOLOv5 |
|----------------------|---|---|
| Architecture | Utilizes a complex architecture with a CSPDarknet53 backbone, which includes multiple convolutional layers and skip connections. | Introduces a simplified architecture that focuses on efficiency and speed. Replaces the CSPDarknet53 backbone with a smaller and more streamlined CSPNet-based architecture. |
| Model Size and Speed | Larger model size, requiring more computational resources. Provides slightly slower inference times compared to YOLOv3. | Smaller model size, achieving faster inference times without compromising much on accuracy. |
| Performance | Aims to improve object detection performance by introducing various techniques, such as PANet, SAM, and PAN. Achieves state-of-the-art accuracy on popular object detection benchmarks. | Aims to improve object detection performance by introducing various techniques, such as PANet, SAM, and PAN. Achieves state-of-the-art accuracy on popular object detection benchmarks. |
| Training Approach | Adopts a traditional single-stage object detection training approach, utilizing anchor boxes and predefined anchor scales. Employs the focal loss function to address class imbalance issues. | Introduces a new approach called “Generalized Anchors,” which adapts anchor sizes during training. Incorporates a focal loss function and implements a novel loss function called “CIoU loss” to improve localization accuracy. |
| Availability | Open-source implementation widely used and supported by the computer vision community. Available as a publicly accessible codebase. | Open-source implementation is gaining popularity due to improved speed and efficiency. Codebase is publicly available, allowing researchers and developers to utilize and contribute to its development. |

Research Gap

Recently, the Internet of Things (IoT) has gained significant popularity as a rapid and effective technique for detecting fires, particularly in the context of forest fire prevention. The IoT forest fire system is designed to facilitate continuous monitoring and detection of forest fires, operating 24/7. This system relies on the strategic deployment of sensors throughout the forest, enabling comprehensive coverage. To establish seamless commu-

nication among these sensors, a low-power wide-area network protocol, such as LoRa, is commonly implemented. Such protocols possess the capability to cover expansive forest areas. Leveraging the data collected by these sensors, combined with advanced machine learning algorithms, early-stage fire detection becomes achievable [34,35]. However, one limitation of IoT is its incapability to directly extinguish fires. In order to overcome this limitation, researchers are investigating the integration of IoT with the robotic ecosystem, which holds promise for forest fire response. In fact, a recent study introduced the IoT-UAV architecture [36], which entails the synergistic collaboration between IoT and Unmanned Aerial Vehicles (UAVs). In this system, the IoT network detects the fire and transmits an approximate location to a UAV, enabling it to locate the precise fire location and perform fire suppression. Utilizing UAVs in forest fire fighting offers several advantages. First, UAVs have the ability to access difficult-to-reach areas, facilitating intervention in challenging terrain. Second, they exhibit a faster response time for deploying fire extinguishing measures. Third, UAVs provide real-time aerial reconnaissance, enabling effective planning of fire suppression strategies. Moreover, their overhead view allows for precise identification of the fire's point of ignition. UAVs can also utilize remote extinguishing systems, reducing risks to human operators. They are capable of dropping water or fire retardant to delay the fire's spread. However, UAVs have certain limitations when it comes to fire suppression. These limitations include limited payload capacity for carrying water or fire retardant, short flight duration, restricted battery life, and difficulties in navigating smoke-filled environments. To overcome these limitations, the collaboration between UAVs and Unmanned Ground Vehicles (UGVs) has gained attention in the field of firefighting forest management [14]. UGVs offer unique advantages that can be leveraged in fire suppression efforts. They excel in navigating rough terrain and reaching remote areas that may be inaccessible for human-operated vehicles. UGVs demonstrate robustness in collision scenarios and can operate effectively in adverse weather conditions and low visibility. Equipped with powerful pumps and larger water tanks, UGVs can hold more water and fire retardant than UAVs, increasing the likelihood of successful fire suppression. Furthermore, UGVs can remain on-site for longer durations, monitoring the fire and preventing re-ignition. They can be outfitted with cameras and sensors to collect valuable data on the fire and the surrounding environment, contributing to future fire management planning. Additionally, UGVs can be remotely operated, minimizing risks to human operators. When combined with UAVs, which provide aerial surveillance and real-time data, coordinated operations between UAVs and UGVs enhance the overall efficiency of the firefighting system.

3. Materials and Methods

In this section, we will dive into the details of the design and implementation of a Cyber-Physical System (CPS) for wildfire detection using ROS-based UAVs (Unmanned Aerial Vehicles) for firefighting and extinguishing. The goal of this CPS is to provide a comprehensive and effective solution for detecting and combating wildfires. The system will use a combination of unmanned systems, each equipped with different sensors and actuators, to gather data and perform various tasks related to firefighting. The UAVs will be responsible for collecting data about the wildfire, including its location, size, and direction of spread. They will also use thermal imaging cameras to identify hotspots and other areas of high temperature. This information will be transmitted to the UGVs, which will use it to plan and execute firefighting strategies. The UGVs will be equipped with various firefighting tools, such as water hoses and foam dispensers, to extinguish the wildfire. They will also be able to navigate through rough terrain, using sensors to detect obstacles and adjust their path accordingly. The CPS will be implemented using Robot Operating System (ROS), a widely used platform for developing robotic systems. This will allow for seamless communication and coordination between the UAVs and UGVs, ensuring that they work together efficiently to achieve the common goal of extinguishing wildfires. Overall, the design and implementation of this CPS will provide a powerful tool for detecting and

combating wildfires, with the potential to save lives and prevent extensive damage to the environment.

3.1. Prerequisite

3.1.1. ROS, Gazebo and PX4

ROS (<https://www.ros.org/>, accessed on 18 April 2023), Gazebo (<https://gazebo.org/home>, accessed on 18 April 2023), and PX4 (<https://px4.io/>, accessed on 18 April 2023) have played a crucial role in the development, simulation, and control of Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs). Together, they have formed a powerful software stack that has enabled the creation, testing, and deployment of advanced robotic systems.

The Robot Operating System (ROS) is a flexible framework that has been specifically designed for robotics applications. It has provided a collection of libraries, tools, and conventions that have facilitated the development of robust and modular robotic systems. With a distributed architecture, ROS has allowed nodes to communicate with each other using a publish–subscribe messaging model. This has simplified the integration of sensors, algorithms, and hardware components, enabling developers to focus on higher-level tasks.

Gazebo has served as a widely used open-source simulator, offering a realistic and physics-based environment for testing and validating robotic systems. It has provided an extensive library of simulated sensors, actuators, and models, enabling developers to create virtual environments that closely resemble real-world scenarios. By supporting high-fidelity simulation of both UAVs and UGVs, Gazebo has allowed researchers and developers to evaluate the performance and behavior of their robots in various conditions without the need for expensive physical prototypes.

PX4, on the other hand, has functioned as an open-source flight control software stack specifically designed for autonomous aerial vehicles. It has provided a comprehensive set of tools and algorithms for controlling UAVs, including autopilot features, navigation, and mission planning. Supporting a wide range of UAV platforms, PX4 has offered interfaces for integrating with various hardware components such as GPS, IMU, and cameras. Its high customizability has made it a popular choice among researchers and commercial drone manufacturers.

Through the synergy of ROS, Gazebo, and PX4, researchers and developers gain access to a powerful and versatile platform for the development, simulation, and control of both UAVs and UGVs. While not obligatory, the utilization of these platforms significantly facilitates the design and testing of complex robotic systems. The ability to evaluate performance and behavior in realistic simulated environments empowers developers to refine their designs before real-world deployment. This approach fosters rapid prototyping, reduces development costs, and enhances the reliability and safety of autonomous systems across various applications, including aerial surveillance, package delivery, autonomous exploration, and industrial automation.

3.1.2. Docker Container

Another crucial technology that has greatly contributed to the development and deployment of modern applications is Docker (<https://www.docker.com/>, accessed on 19 April 2023). Docker has revolutionized the software development process by providing a powerful framework for creating, sharing, and running applications using containerization. One of the key advantages of Docker lies in its efficient virtualization capabilities, allowing for the isolation and streamlined management of applications. The architecture of Docker revolves around a client-server model, where the Docker client interacts with the Docker daemon. The Docker daemon is responsible for essential tasks such as container building, execution, and overall management. Leveraging Docker's capabilities, our CPS Wildfire Detection System has achieved remarkable flexibility and modularity in deploying and managing a diverse range of services. These services include RabbitMQ (<https://www.rabbitmq.com/>, accessed on 19 April 2023), Node-RED (<https://nodered.org/>, accessed

on 19 April 2023), and Nuclio Serverless (<https://nuclio.io/>, accessed on 19 April 2023), among others. By harnessing the power of Docker, we have created an efficient and modular environment that enables seamless integration and deployment of various components within our CPS Wildfire Detection System. Docker's virtualization efficiency allows us to isolate individual applications and manage them effectively. This ensures that each component operates independently while being part of the larger system, promoting scalability, reliability, and easy maintenance.

RabbitMQ

Fundamental to the system has been the RabbitMQ broker, which has implemented the publish/subscribe mechanism, including Message Queuing Telemetry Transport (MQTT), enabling communication between components through message exchange. RabbitMQ has served as a reliable, scalable, and flexible message broker, capable of efficiently handling a high volume of real-time messages. With the utilization of MQTT, a lightweight messaging protocol designed for resource-constrained devices such as IoT devices, RabbitMQ has ensured efficient and effective communication among the system's components.

Node-RED

An important tool that has aided in building the dashboard for the CPS Wildfire Detection System has been Node-RED. It is a visual programming tool that has facilitated the creation of flow-based applications through a user-friendly drag-and-drop interface. Node-RED has offered a web-based platform for developing and deploying IoT applications. Furthermore, it has supported various input and output protocols, including MQTT, HTTP, and TCP/IP. The platform has come equipped with a vast library of pre-built nodes that have been utilized to construct intricate flows. Additionally, it has been employed to create a flow-based application that has established a communication socket, allowing for message exchange between ROS-based and MQTT-based devices.

Nuclio Serverless

Finally, another important aspect has been Nuclio Serverless, a cloud-native platform that has streamlined the development and deployment of serverless-based applications, including notification systems. It has offered several features, such as integration with services such as IFTTT (<https://ifttt.com/>, accessed on 2 April 2023) and the Node-RED framework. This integration has provided an effortless way to create robust notification systems, connecting them with email and/or chatbot functionalities. These systems have been crucial for informing firefighter operators and establishing a connection between the CPS Wildfire Detection System and an external monitoring and control room.

3.2. YOLO Object Detection

The accurate identification and localization of fire incidents in real time are crucial for effective fire detection systems. To achieve this, our unified Unmanned Ground Vehicle (UGV) and Unmanned Aerial Vehicle (UAV) system leverages the You Only Look Once (YOLO) family of object detection algorithms. YOLO algorithms have demonstrated exceptional performance and efficiency in the field of computer vision.

3.2.1. YOLOv4: Object Detection for UGV Component

In the UGV component of our system, we implement YOLOv4 as the primary object detection algorithm. YOLOv4 is renowned for its state-of-the-art accuracy and its compatibility with the darknet framework and Robot Operating System (ROS). This compatibility enables seamless integration within the UGV system, leveraging the darknet-ROS package. Furthermore, considering the resource limitations of the system, we choose to utilize YOLOv4-Tiny, a compressed version designed for CPU-enabled devices. YOLOv4-Tiny achieves high-speed object detection with a reduced number of convolutional layers and YOLO layers, along with fewer anchor boxes for prediction.

3.2.2. YOLOv5: Object Detection for UAV Component

To extend the capabilities of our system, we incorporate YOLOv5 into the UAV component. YOLOv5 offers a streamlined architecture and optimized data augmentation strategies, making it ideal for resource-constrained UAVs. Leveraging its efficiency, YOLOv5 empowers the UAV to perform real-time fire detection and localization, enhancing aerial surveillance and coordination with the UGV. The integration of YOLOv5 within the UAV component showcases its advantages in terms of inference speed and its ability to handle resource limitations.

By harnessing the strengths of YOLOv4 in the UGV component and YOLOv5 in the UAV component, our unified system achieves robust object detection capabilities and seamless integration within the overall framework. Moreover, the collaboration between UGVs and UAVs, facilitated by YOLO algorithms, enables real-time fire detection and localization, providing an effective solution for forest fire monitoring and detection.

3.2.3. Implementation of YOLO in ROS

The integration of the You Only Look Once (YOLO) object detection algorithm with the Robot Operating System (ROS) plays a pivotal role in enabling fire detection and localization on our robotic platforms. To facilitate this integration, we utilize the widely adopted “darknet_ros” package, which leverages the “darknet” framework—a neural network framework developed in C and CUDA.

The “darknet_ros” package provides a seamless ROS interface for utilizing the YOLOv4 object detection algorithm. Specifically, we utilize the yolov4-for-darknet_ros package within the “darknet_ros” package. To make use of this package effectively, we provide two key files: the configuration file (.cfg) and the tiny-weights file (.weights) for the pre-trained YOLOv4-Tiny model.

- **Configuration File (CGF):** This file contains essential information regarding the network architecture and hyperparameters used during both training and testing phases, by configuring these parameters within the YOLOv4-Tiny model in ROS, we optimize the object detection capabilities of our robotic platforms, facilitating efficient fire detection and localization.
- **Weights:** This file is a binary file that contains the learned parameters of the neural network model. This file is utilized to make predictions on new data by applying the knowledge gained from the training process.

In addition to the ‘darknet_ros’ package, we also require another ROS package called ‘darknet_3d_ros’ to enable fire localization relative to the camera. This is crucial in this context because we aim to maintain a safe distance from the fire and position the pump in the correct direction for extinguishing it. The ‘darknet_3d_ros’ package provides the integration of 3D object detection using YOLO, which uses the RealSense Depth Camera of the Jackal Rover to capture 3D point cloud data and perform fire localization. The main output of the package darknet_3d_ros package is the topic /darknet_ros_3d/markers. The ROS topic /darknet_ros_3d/markers contains visualization markers that represent the position and orientation of fires detected by the YOLOv4-Tiny model in the x, y, and z coordinates. The markers provide a way to visually represent the position, size, and orientation of the detected objects in 3D space. The markers are represented as a set of rectangular cuboids, where each cuboid represents a detected object. The cuboids are positioned and sized according to the 3D bounding box of the object in the scene.

3.3. Cyber-Physical System for Wildfire Detection and Firefighting

The concept of the CPS, originally proposed by NASA for unmanned space exploration, extended to the military environment to reduce military casualties by allowing soldiers to remotely control weapons from a command post. Nowadays, CPS has been widely adopted in critical sectors such as transportation, energy, healthcare, and manufacturing as part of *Industry 4.0*, where human-machine Interfaces are widely utilized to enhance production

process performance while simultaneously decreasing the occurrence of emergencies and accidents [37,38]. The CPS is an extension of control systems and embedded systems and has evolved rapidly due to advancements in cloud computing, sensing technology, communication, and intelligent control. The fundamental principle of the CPS involves integrating cyber processes such as sensing, communication, computation, and control into physical devices to monitor and manage the physical world. In the context of Cyber-Physical Systems (CPSs), effective communication is essential for facilitating seamless coordination and integration among various components. Machine-to-Machine (M2M) communication stands out as a significant communication paradigm within CPS. M2M communication specifically refers to the direct exchange of data between sensors, machines, robots, and other devices without requiring human intervention. After taking into account the previous considerations and current research on forest prevention, a new solution has emerged in the form of a CPS designed for wildfire detection and firefighting. This innovative design aims to provide firefighters with a tool that can better support their daily activities and prevent fires at an early stage. The CPS for wildfire detection and firefighting will incorporate advanced technologies such as machine learning, sensors, and autonomous systems. These components will work together to detect and fight wildfires in real time, allowing firefighters to respond quickly and effectively. The system will also provide valuable data and insights that can be used to improve future prevention efforts. For example, the data collected can be analyzed to identify areas that are at high risk for wildfires, and preventive measures can be taken to reduce the likelihood of fires starting in those areas. Overall, the CPS for wildfire detection and firefighting could potentially represent a significant step forward in the fight against wildfires. By leveraging the latest technologies and research, firefighters will be better equipped to prevent and respond to fires, ultimately helping to protect communities and preserve natural habitats. In the upcoming section, there will be a comprehensive overview of the methods and tools used to design and develop cyber-physical systems for wildfire detection and firefighting.

Architecture

The proposed architecture (Figure 2) is composed of different heterogeneous and distributed nodes, each of them serving a specific purpose:

Communication Sensor: The communication between the sensors and the robots in our project is facilitated by the Robot Operating System (ROS) framework, which provides a flexible and robust infrastructure for building robotic systems. To enable data exchange between different components of the system, we utilize ROS topics, which employ a publish–subscribe mechanism. In our system, we employ a Jackal Rover as one of the robots, equipped with sensors such as the fire detection system powered by You Only Look Once (YOLO). These sensors communicate their data to other components using ROS topics. When a fire is detected, relevant information, including the status of the fire (extinguished or not), is published on a specific ROS topic. This enables other components, such as the Unmanned Aerial Vehicle (UAV) and the ground control station, to subscribe to the topic and receive the necessary data, facilitating coordinated actions.

Communication Unmanned Devices: To establish communication between the UAV and the UGV, we utilize the ROS-Bridge package (http://wiki.ros.org/rosbridge_suite, accessed on 28 April 2023). ROS-Bridge provides a WebSocket interface to ROS topics and services, facilitating the exchange of ROS messages over the internet using the Wi-Fi antenna equipped on the devices. By running the ROS-Bridge on a computer or embedded system acting as the ground control station, we establish a communication protocol between the UAV and the UGV.

In our implementation, the ROS-Bridge launcher file is configured with the IP address of the ground control station, which serves as the communication endpoint. Once the ROS-Bridge server is launched on the ground control station machine, data flow

between the UAV and UGV is facilitated through the ROS-Bridge, with the ground control station acting as an intermediary.

For example, if the UAV needs to communicate GPS data to the UGV, the ground control station receives this data from the UAV via the ROS-Bridge WebSocket. Subsequently, the data are forwarded to the UGV, ensuring seamless coordination between the two robots. This communication process is also employed to establish communication between unmanned devices and the IoT device. ROS-Bridge enables the communication between both ROS-enabled and non-ROS-enabled devices, enabling wireless communication facilitated by the use of Wi-Fi technology, which is also equipped on the IoT device.

Wildfire Detection Node: This is a prototype based on a low-power system of chip microcontroller (ESP32 Wroom 32) and equipped with multiple sensors such as temperature and humidity sensors, an infrared and a CO₂ sensor. The prototype uses a GPS module for fire localization and an IP ESP32 Camera that is linked to a real-time fire detection system based on YOLOv5. This integration enables the device to quickly and accurately detect the presence of a wildfire. The combination of these different sensors and technologies creates an advanced and all-encompassing solution for monitoring and detecting forest fires.

Unmanned Ground Vehicle (UGV): This is an ROS-based Jackal Rover designed specifically for wildfire detection and extinguishing. Equipped with the YOLOv4-Tiny CNN model, it can search for fires after receiving the approximate location from the IoT node or the more precise location from the UAV device. The UGV can also detect the location of the fire by combining the YOLOv4-Tiny CNN model and Realsense depth cameras, allowing it to maintain a safe distance and direct the pump for extinguishing the fire. The UGV operates autonomously by following a navigation path and responding immediately without human intervention. Additionally, it is equipped with Lidar technology, enabling a collision avoidance mechanism to navigate around obstacles and avoid collisions. Table 2 provides the Capability Matrix defined in [39] and represents a summary of the UGV's technical details, encompassing its microcontroller, processors, sensors, actuators, mobility, and communication. These specifications ensure the UGV's safe operation during autonomous missions.

Table 2. UGV Technical Details.

| Robot Name | Microcontroller/ Processors/OS | Sensors/Actuators | Mobility | Communication |
|---------------|--|--|------------------|------------------|
| <i>Jackal</i> | Raspberry Pi 3 Model B with CPU 1.2 GHz Quad Core, 1 GB RAM, ROS Noetic | IMU, Wheel Encoders, GPS, Lidar UTM-30LX, Intel RealSense D435 camera | 4 × 4 drivetrain | Bluetooth, WI-FI |

Unmanned Aerial Vehicle (UAV): This is an ROS-based UAV designed specifically for wildfire detection and fire extinguishing. Equipped with a fireball-extinguishing payload, this UAV is capable of quickly and efficiently extinguishing fires before they can cause significant damage. The UAV fireball extinguisher is released automatically when the fire is detected, thanks to a YOLOv5 algorithm wrapped in an ROS package and starter as an additional ROS node. This allows the UAV to quickly search for fires by following a navigation path and responding immediately, without any need for human intervention. To ensure safe operation and autonomous mission, the UAV is also equipped with three front-facing RealSense depth cameras, one on the left, one on the right, and one in the center, which provide 3D depth information to enable the collision avoidance mechanism. This ensures that the UAV can navigate around obstacles and avoid collisions, even in challenging environments. Additionally, a bottom Realsense camera is used in the UAV model, and this camera allows the UAV to accurately detect fires in forests or uneven terrain. Table 3 provides the

Capability Matrix defined in [39] and represents computational capabilities required by Unmanned Aerial Vehicles (UAVs) to support Ubuntu 20.04 with ROS Noetic, the algorithms for communication between the central unit and the onboard PX4 Autopilot, but they also represent the minimal capabilities necessary to enable flight control and real-time image recognition performed by the onboard sensor and the YOLOv5 model running onboard the UAV.

Table 3. UAV technical details.

| Robot Name | Microcontroller/ Processors/OS | Sensors/Actuators | Mobility | Communication |
|------------|---|--|------------|------------------|
| 3DR Iris | Cortex M4 168 Mhz/256 KB RAM/2 MB Flash, ROS Noetic, PX4-Autopilot | IMU, GPS, Intel RealSense D435 camera | quadcopter | Bluetooth, WI-FI |

Ground Control Station: This is a system made up of three dashboards that allow real-time monitoring of the forest, UAV onboard status, and telemetry, as well as the UGV. This system uses the Node-RED framework to create an ROS-based communication bridge between various autonomous systems, enabling efficient and fast transmission and reception of information. Furthermore, the ground control station acts as a RabbitMQ broker, meaning it can receive data streams from the wildfire detection system through the MQTT protocol. This allows for a comprehensive overview of the forest conditions and enables prompt action in case of emergencies. In addition, the ground control station uses third-party software such as QGroundControl, which is a user interface for PX4. QGroundControl can configure and operate various types of vehicles, including copters, airplanes, rovers, boats, and submarines, as well as different types of firmware such as PX4. This makes the ground control station a highly versatile system suitable for multiple applications.

Notification System Node: This is a system that generates notifications if a fire is detected. It is based on a serverless function and provides a simple and efficient solution for promptly notifying the competent authorities in case of an emergency. Furthermore, the notification node uses an instant messaging service and a chatbot to forward information about the location or area where the fire has been detected. This allows for detailed and precise information to be provided to the competent authorities for a quick response and efficient management of the emergency.

As we can see in Figure 2, heterogeneous devices communicate and collaborate with each other to accomplish specific tasks, supported by a publish–subscribe mechanism to exchange information through topics. The next section will describe all the frameworks used for the development of our architecture. We will provide a thorough explanation of each node, what data it processes, the important code snippet and algorithm, and how it interacts with other nodes and have a complete overview of the architecture.

3.4. Wildfire Detection Node

The wildfire detection node is designed to be a wireless edge device that is installed in a forest to detect flames in real time using object detection techniques. In addition to its flame detection capability, it also includes several other sensors that can provide additional environmental data. These sensors include a temperature and humidity sensor, geolocation devices, a CO₂ sensor, and an Infrared (IR) sensor. The temperature and humidity sensor measures the weather in the forest, which can be used to identify potential fire hazards. Additionally, the geolocation devices can provide the location of the node within the forest, which can be used to track the spread of a fire. Moreover, the CO₂ sensor detects the presence of carbon dioxide, which can be an indicator of combustion. The IR sensors can detect heat signatures, which can be used to detect flames even in low-light conditions.

Together, these sensors can provide valuable data that can be used to detect wildfires early, respond quickly, and potentially prevent the fire from spreading. The wireless nature of the node also means that it can be deployed in remote areas and communicate its data back to a central hub or control room in real time, making it a powerful tool for wildfire detection and prevention.

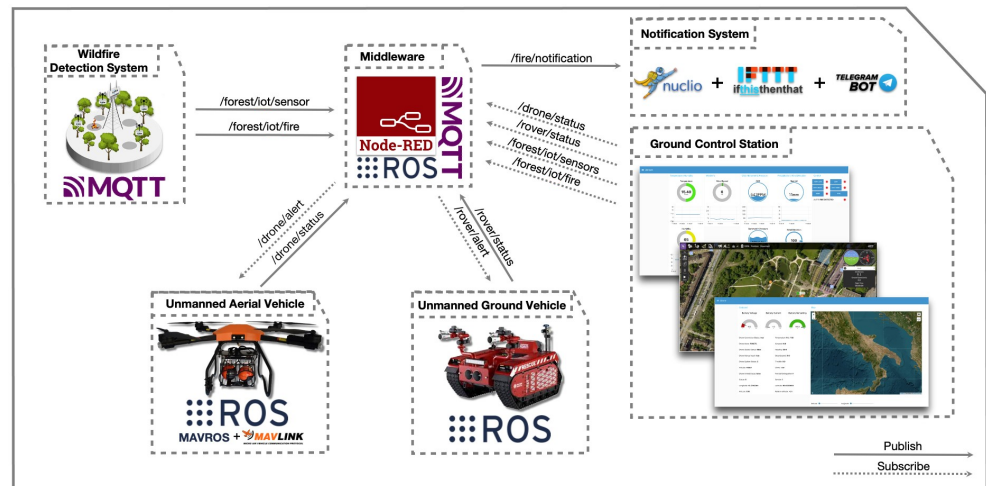


Figure 2. Cyber-physical system for wildfire detection and firefighting.

Circuit Diagrams

Figures 3 and 4 show the circuit diagram and the schematic for the wildfire detection system. In the following paragraph, a brief overview of the components will be provided to enhance comprehension of the pinout, sensor function, the application context, and how these sensors can give help in early fire detection.

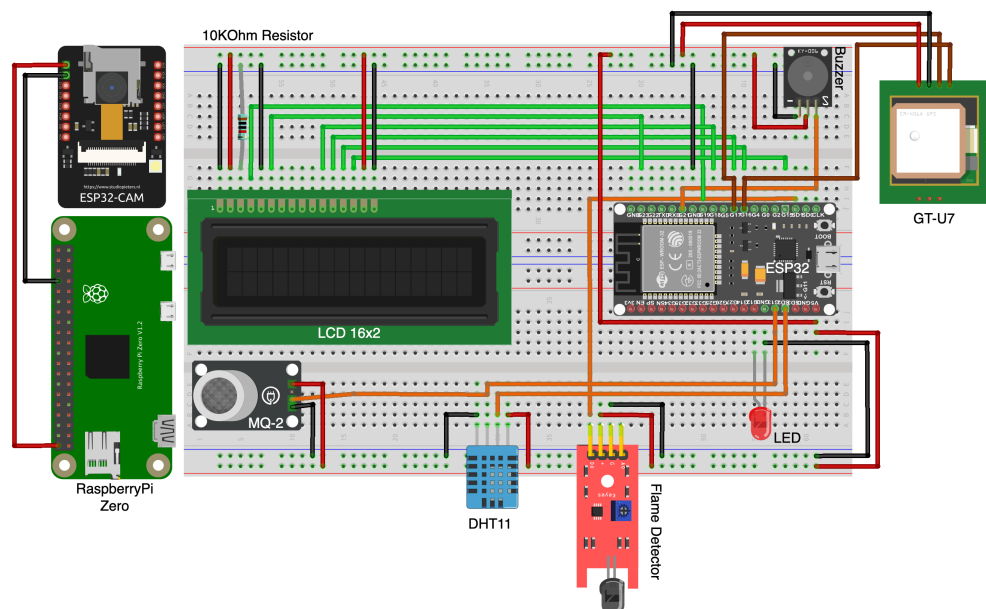


Figure 3. The circuit diagram designed using Fritzing to display the pin configuration.

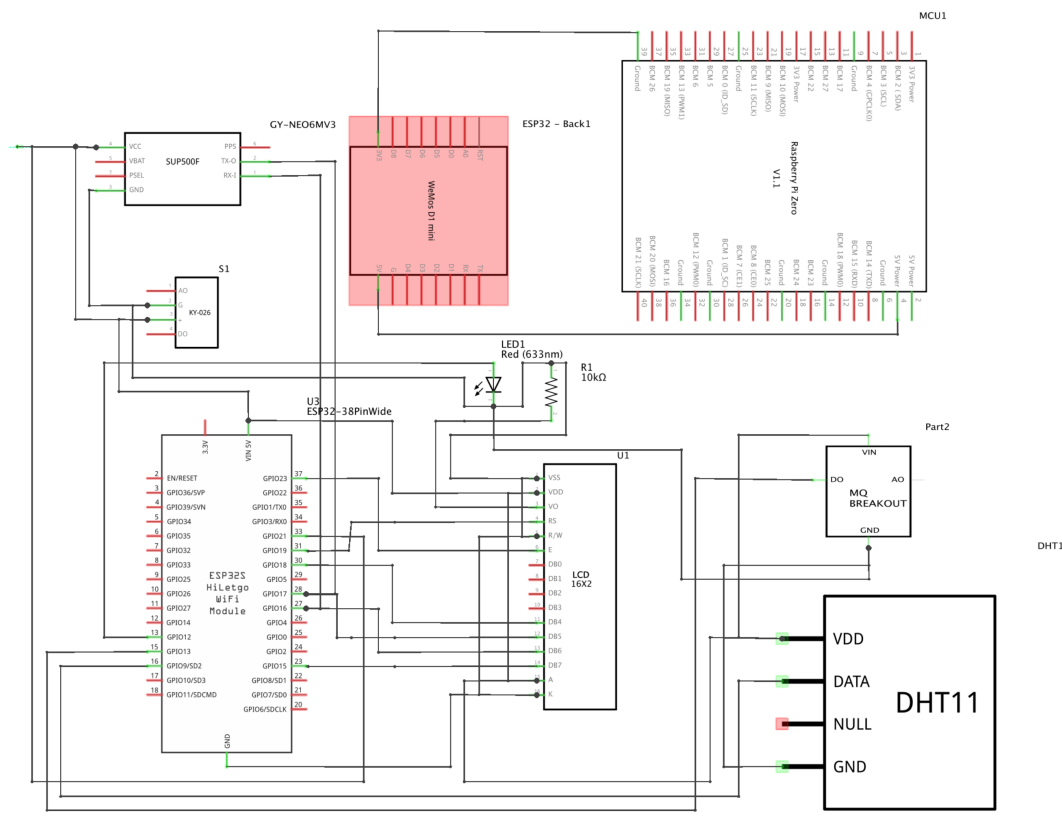


Figure 4. Schematic of the Wildfire Detection Node (green pin connected, red pin not connected).

3.5. Ground Control Station Node

The ground control station allows the monitoring of both the forest and autonomous systems via a custom dashboard. It also facilitates communication between MQTT-based and ROS-based devices employing specific flows that act as a communication bridge. The system also processes data series and provides real-time insights, allowing the operator to make decisions and take appropriate action as required. In the next section, we will be discussing the primary setup required to allow for MQTT communication between IoT devices.

3.5.1. Node-RED Forest Dashboard

Node-RED Dashboard is an extension to Node-RED that provides web-based dashboards for visualizing data and controlling devices. The dashboard created is used to monitor the status of the wildfire detection system, and provide real-time information about temperature, humidity, and wind speed (Figure 5).

Additionally, it is used to connect the dashboard to different MQTT topics for data transmission and fire notification. Furthermore, other flows are used to create an MQTT to ROS communication bridge between the wildfire detection system and the UAV (Figure 6), but also between the UAV and the notification system (Figure 7).

3.5.2. Node-RED UAV Dashboard

Besides the dashboard used for monitoring the forest, an additional dashboard has been developed to monitor the onboard telemetry of the UAV and its real-time position on a map (Figure 8). The dashboard is managed by a flow of nodes subscribed to various topics. The real-time data stream generated by the UAV is redirected to the nodes for graphical display. Different nodes have been developed in order to obtain the necessary information generated by the UAV. As shown in Figure 9, the nodes receive all the information from the MAVROS topics, which, as described previously, is the middleware that enables the com-

munication between the PX4 Autopilot used by the UAV to fly and the ROS environment installed on the companion computer for the collision algorithm.

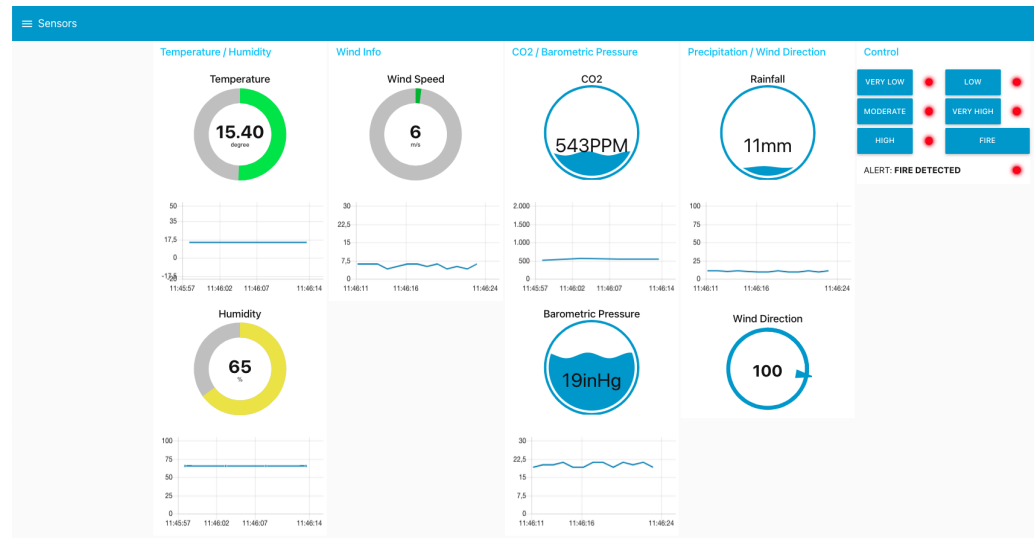


Figure 5. Forest monitoring dashboard.

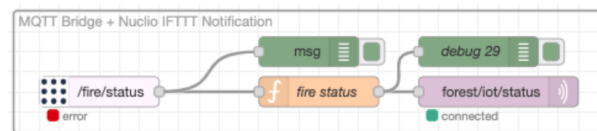


Figure 6. Communication bridge: UAV to notification system.

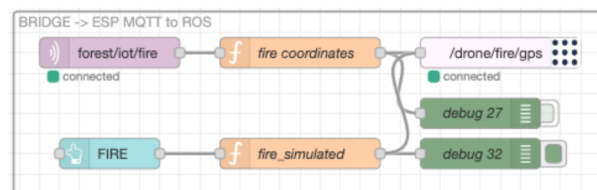


Figure 7. Communication bridge: wildfire detection system to UAV.

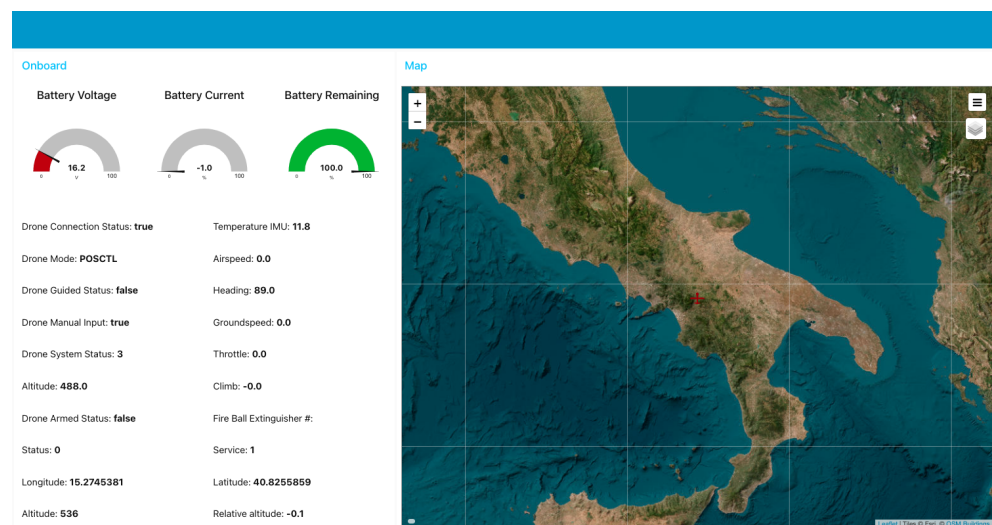


Figure 8. UAV monitoring dashboard.

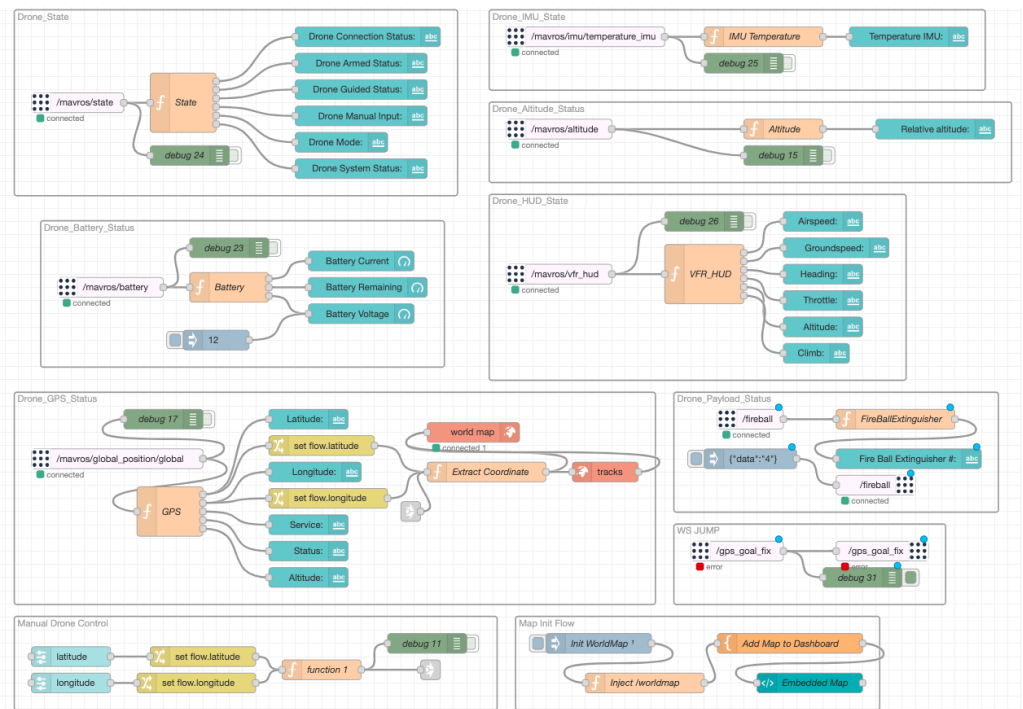


Figure 9. Node-RED flow for the UAV monitoring dashboard.

3.6. Notification Systems Node

The Notification Systems node comprises two serverless functions managed through the Nuclio platform. The two functions are capable of sending a fire alert to all connected devices, such as the UAV and wildfire detection system. The system is also connected to an IFTTT web-based service that uses a Webhook for HTTP requests to a specified URL with parameters and authentication details. The combination of webhooks and IFTTT allows the automation of various tasks and the creation of custom workflows. In our case, it is able to send messages to a Telegram chatbot. This system is vital because it provides real-time alerts to the firefighter operator, ensuring that they are constantly informed and able to respond quickly. The scalability of the Notification System allows an easy way to upgrade the services and the automated task, but also the flexibility to be integrated into other existing alerting systems.

Nuclio Serverless Function

The following code snippet Listing 1 is an example of the JavaScript functions handler managed by the Nuclio platform. The functions are characterized by an event trigger that takes two parameters: *context* and *event*. The event parameter contains data that the function needs to execute. The process first parses the JSON data in the event body, then checks if the sensor data indicates a fire, and if so, it sends a notification to the IFTTT web service that automatically generates a message for the Telegram chatbot. The notification messages include the sensor value and the geographic value of the fire position.

Listing 1. ForestHandler serverless function.

```

...
exports.handler = function(context, event) {
var forestJson = JSON.parse(event.body);
  if (forestJson.sensor == "FIRE_ON") {
    for (let i = 0; i < 1; i++){
      rest.post(
        'https://maker.ifttt.com/trigger/fire_notification/with/key/' +
        ifttt_event_key, {data: {value1: forestJson.sensor,

```

```

        value2: forestJson.position[0],
        value3: forestJson.position[1]}}).on('complete',
        function(data){console.log("Forest status: "+forestJson.sensor+
        "Latitude: " + forestJson.position[0] +
        "Longitude: " + forestJson.position[1]);});
    }
}
context.callback("");
};
...

```

3.7. ROS-Based UAV Node

The UAV node is a crucial component of our cyber-physical system for wildfire detection and firefighting. As previously mentioned, this node is an integrated system composed of an ROS/MAVROS environment installed on the companion computer that is responsible for processing the data acquired by the UAV's sensors, including cameras and other environmental sensors. It also provides a platform for implementing advanced algorithms for processing sensor data and for controlling the UAV's behavior. The PX4 Autopilot, on the other hand, is responsible for generating the control signals needed to navigate the UAV to specific locations in the wildfire-affected area. The use of MAVROS as a middleware enables seamless communication between the ROS environment and the PX4 Autopilot, allowing for the real-time exchange of sensor data and control signals. This communication is critical for ensuring that the UAV operates safely and effectively, and it is essential for implementing advanced algorithms for wildfire detection, firefighting, and collaboration with all the other distributed systems that are involved in the architecture.

UAV ROS Launcher

In order to test the behavior of the UAV, how to detect fire spots, and how to avoid a collision, a 3D simulation environment has been used. The setup of the simulated UAV environment consists of the following ROS launcher:

- **Local Planner Launcher:** This is used for launching a simulation environment for a UAV with triple-depth cameras and a local planner nodelet
- **UAV Controller Launcher:** This provides functions to set different modes of operation for the PX4 Flight Controller Unit. It includes functions for takeoff, landing, arming, and disarming the FCU, as well as for setting different flight modes such as AUTO, STABILIZED, OFFBOARD, ALTCTL, and POSCTL modes. These functions use ROS and MAVROS service proxies to call FCU commands and change the mode of the UAV.
- **YOLOv5 Detection Launcher:** This specifies the detection configuration parameters for the YOLOv5 model, including the weights and data paths, confidence and IoU thresholds, the maximum number of detections, a device used for inference, and line thickness for bounding box visualization. The launch file also allows for setting the input image topic and output topic for the YOLOv5 detection node, as well as the option to publish annotated images and visualize the results using an OpenCV window.
- **ROS-Bridge Launcher:** This is a protocol that enables bidirectional communication between a client and a server over a single TCP connection. It allows real-time communication without the need for the client to constantly poll the server for new data. WebSockets are commonly used in web development for real-time web applications, but they can also be used for other types of applications, such as Node-RED. ROS-Bridge uses JavaScript Object Notation (JSON) to encode and decode ROS messages that are sent and received over the WebSocket connection.

3.8. ROS-Based UGV Node

The UGV node plays a critical role in our cyber-physical system for wildfire detection and firefighting. As described earlier, this node serves as an integrated system that consists of a companion computer with an ROS environment. The ROS environment is responsible for processing data obtained from various sensors on the UGV, including cameras and environmental sensors. In our implementation, we leverage several key ROS packages to enhance the capabilities of the UGV. Firstly, we integrate YOLOv4, a state-of-the-art object detection algorithm, for fire detection. This enables the UGV to identify and locate fires accurately. Secondly, we utilize the Move Base package to implement navigation planning algorithms, which allow the UGV to navigate safely and effectively while avoiding obstacles in its environment. This capability is crucial for ensuring the UGV's operational efficiency during firefighting missions. Additionally, we employ GMapping, an ROS package for mapping, to enhance the situational awareness [40] of the UGV. By constructing a map of the environment, the UGV can provide valuable information to the ground station operators, aiding them in making informed decisions.

UGV ROS Launcher

In order to test the behavior of the UGV, how to detect fire spots, and how to avoid a collision, a 3D simulation environment has been used. The setup of the simulated UGV environment consists of the following ROS launcher:

- **Robot Localization Launcher:** Despite incorporating a GPS sensor, relying solely on GPS data for outdoor localization in the Jackal Rover poses challenges. The imprecision of GPS measurements leads to significant fluctuations in values, and factors such as slippage and uneven surfaces can cause deviation from the GPS trajectory. Moreover, GPS only provides position information without orientation details. To address this, integrating a compass such as an Inertial Measurement Unit (IMU) becomes essential. Fusing data from multiple sensors, including GPS, IMU, and wheel odometry, mitigates noise and errors, improving the reliability of GPS localization.
- **Path Planning:** Once we have implemented GPS localization on our Jackal Rover, we can proceed with fulfilling a key requirement of our project, which is forest navigation. This task is crucial as the robot needs to reach the GPS location of the fire while avoiding obstacles. To accomplish this, we have opted for the Navigation Stack, a collection of ROS nodes and algorithms for autonomous robot navigation. Our project incorporates the Move_Base node, the core component of the Navigation Stack, which guides the robot to its goal while considering obstacles. We have created a launch file, `move_base.launch`, to configure the Navigation Stack. This launch file loads various parameters from `.yaml` files for different components, including the costmap, local planner, and `move_base` itself.
- **Mapping launcher:** Mapping is a critical component of implementing Unmanned Ground Vehicles (UGVs) with ROS. It allows the Jackal Rover to create a detailed representation of the environment it traverses, providing valuable information to firefighters. Simultaneous Localization and Mapping (SLAM) is employed to build a map while tracking the robot's location. Among the available 2D SLAM algorithms in ROS, GMapping is selected for its particle filter approach and utilization of odometry and LIDAR scans. The GMapping algorithm is implemented through the GMapping ROS package, utilizing the `slam_gmapping` node to create a 2D map as the robot moves. This efficient and accurate mapping solution enhances situational awareness for firefighting operations.
- **Fire Approaching Launcher:** To achieve fire detection and extinguishing, an ROS-based control logic system is implemented for the UGV. This system facilitates coordination among various ROS nodes responsible for GPS localization, obstacle avoidance, mapping, fire detection, and localization. The control logic, implemented as Python code, acts as the robot's brain, enabling collaboration between these nodes. GPS

coordinates of the fire are received, and a series of waypoints is generated to cover a wider detection area. Custom ROS nodes handle navigation and fire detection, utilizing algorithms and neural networks. The robot incrementally approaches the fire, maintains a safe distance, extinguishes it, and returns to the base. This control logic ensures efficient and effective firefighting operations for the UGV.

4. Simulation

This section presents the results achieved by simulating the CPS system for wildfire detection and extinguishing. Through frameworks such as ROS, Gazebo, and RViz, it was possible to simulate, test, and continuously develop the system. By partially incorporating hardware, such as IoT sensors, a Software-in-the-Loop system was created, allowing for seamless integration between software and physical components. This approach enables a comprehensive evaluation of the system's performance and functionality, ensuring its effectiveness in real-world scenarios and providing a graphical representation of the mission that the UAV and UGV will conduct after the notification of the fire from the wildfire detection system.

4.1. Unmanned Aerial Vehicle

The UAV's behavior has been extensively tested and validated through simulations conducted in Gazebo, a popular robot simulation environment. The simulations were performed on a machine with an Intel Core i7 2.8 GHz Quad-Core CPU and AMD Radeon R9 M370X 2 GB, running Ubuntu 20.04 LTS with ROS version Noetic. These specifications ensured sufficient computational power and compatibility with the ROS framework for accurate simulation results. In the simulations, we created a realistic forest environment, incorporating vegetation and fire representation. The fire spots were represented by a cube, while trees acted as obstacles that the UAV had to navigate around. To evaluate the UAV's performance, we conducted a series of tests:

- **Test 1. UAV Navigation:** The first test, we can see how the UAV, after receiving its mission, begins its navigation by following the path as shown in Figure 10. Then, the UAV passes from one waypoint to another in search of the fire. If the fire is not found during the first generated navigation mission, the system, based on the initially received coordinates, generates a new navigation path with greater coverage, a higher number of waypoints, and a higher altitude. This is done in order to increase the probability to detect the fire.

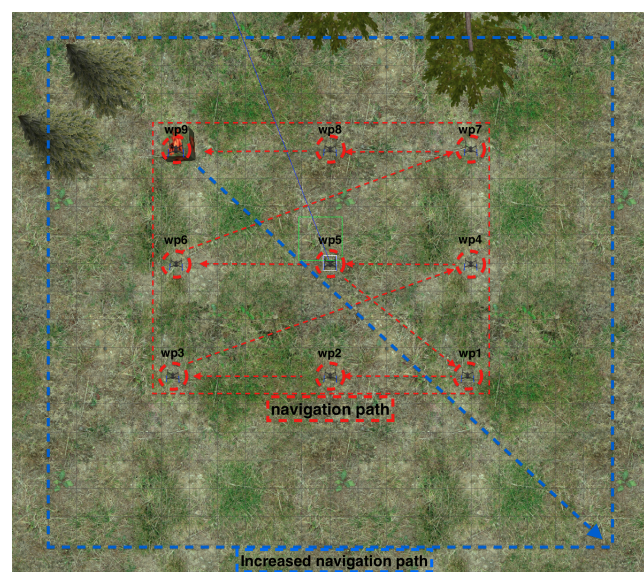


Figure 10. UAV waypoints and navigation path.

Figure 11 depicts the sequence of actions taken by the UAV to initiate and carry out a mission autonomously. Once the wildfire detection system notifies the UAV of a fire, it will *arm* the motor, switch to *offboard mode*, and navigate through the waypoints. If no fire is detected in the initial zone, it will increase its coverage range (Figure 11). However, if it detects a fire, it will return to its *home base and land*.

```

gerardomartino@ubuntu: ~
gerardo... x gerardo... x gerardo... x gerardo... x gerardo... x
^Cgerardomartino@ubuntu:~$ rosrund local_planner goto_latlong_v2.4.py
Arming Succeeded
Change in OFFBOARD
Changed to Offboard Mode
WP: 1
Fire Detection at coordinates: 40.8255412 / 15.2744787
WP: 2
Fire Detection at coordinates: 40.8255861 / 15.2744787
WP: 3
Fire Detection at coordinates: 40.825631 / 15.2744787
WP: 4
Fire Detection at coordinates: 40.8255412 / 15.2745381
WP: 5
Fire Detection at coordinates: 40.8255861 / 15.2745381
WP: 6
Fire Detection at coordinates: 40.825631 / 15.2745381
WP: 7
Fire Detection at coordinates: 40.8255412 / 15.2745975
WP: 8
Fire Detection at coordinates: 40.8255861 / 15.2745975
WP: 9
Fire Detection at coordinates: 40.825631 / 15.2745975
Fire Detection Increase Range
^Cgerardomartino@ubuntu:~$
    
```

Figure 11. UAV waypoint navigation based on GPS coordinate.

- **Test 2. UAV Fire Detection:** The second test, the scenario depicted in Figure 12, shows how the UAV on a mission identifies the fire approximately at waypoint three. Then, a simulation of a fireball extinguisher release takes place, followed by sending the coordinates to the UGV node. Finally, the UAV returns to its home base.

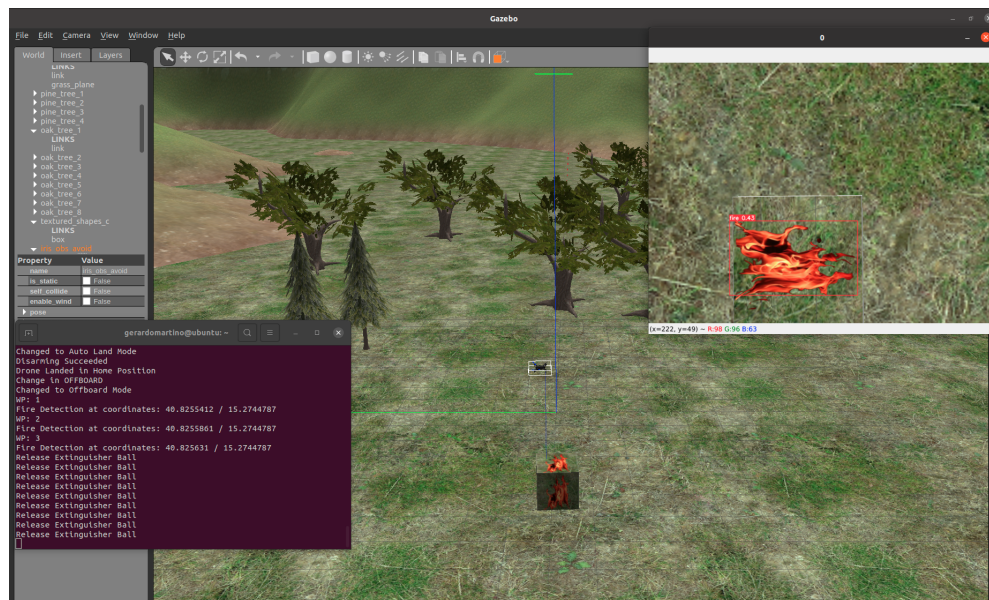


Figure 12. UAV fire detection at Waypoint #3.

- **Test 3. UAV Obstacle Avoidance:** The third test represents the simulation results for testing a collision avoidance system, using the 3DVFH+ algorithm in a 3D environment. Figure 13 illustrates the movement of the UAV in the presence of obstacles. While RViz only displays trees and not the terrain, the algorithm takes into account altitude changes, ground characteristics such as differences between grass and rock, and the physics of all objects during landing. RViz is a useful tool to observe the algorithm's

computed path (red line) and the UAV's actual path (green dotted line) in real time. However, the algorithm only considers obstacle detection and collision avoidance parameters and not the UAV's flight parameters. Therefore, the autopilot must manage the collision avoidance system commands and combine them with vehicle dynamics. This results in differences between the two paths, especially in turns or changes of trajectory. If the vehicle is unable to complete the mission, the path is changed, or the collision avoidance system's parameters are varied while maintaining the same mission.

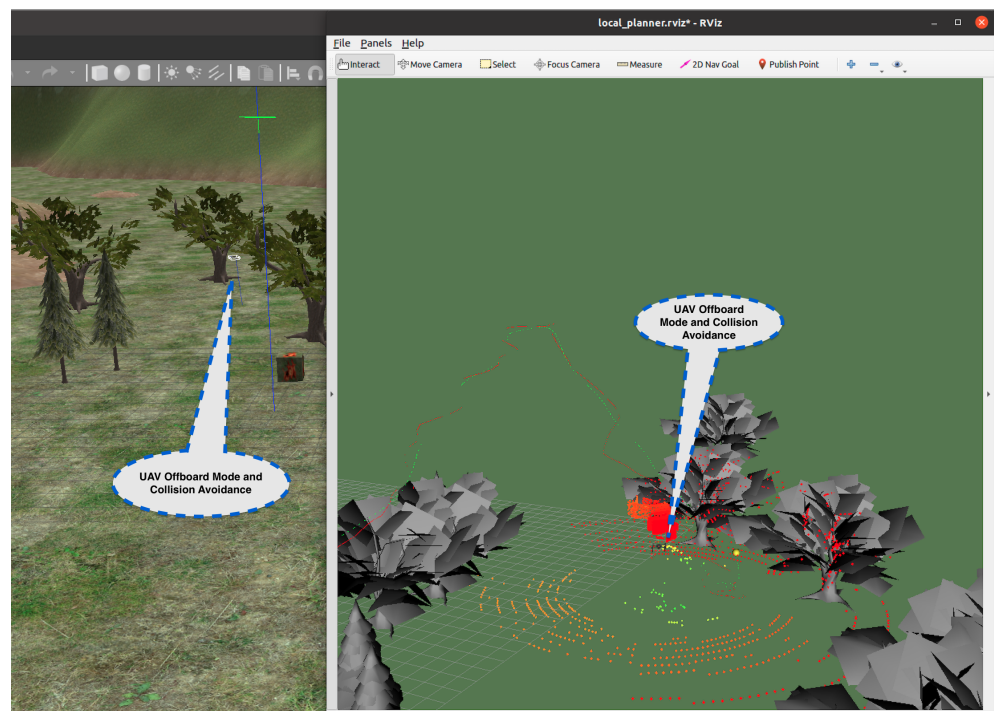


Figure 13. UAV collision avoidance using 3DVFH+ algorithm.

Throughout all the tests, the UAV demonstrated its ability to autonomously locate and suppress fires, successfully overcoming the obstacles posed by the simulated trees. Additionally, the Unmanned Aerial Vehicle (UAV) acts as an aerial support platform, providing valuable assistance in fire detection and suppression operations. The UAV offers several advantages and capabilities that complement those of the UGV. Firstly, the UAV enhances situational awareness by providing a bird's-eye view of the fire. Secondly, the UAV enables intervention capabilities in scenarios where the UGV may face limitations or challenges. For example, if the UGV encounters inaccessible terrains or areas with unfavorable conditions, the UAV can be deployed to reach those locations and perform necessary actions to control or suppress the fire. By incorporating the UAV into the system alongside the UGV and the wildfire detection systems, our solution addresses a broader range of fire scenarios in diverse forest environments. The synergy between the UGV and UAV optimizes the effectiveness and efficiency of fire suppression operations, leveraging the unique capabilities of each platform to tackle different challenges and enhance overall performance. However, it is important to note some limitations and capabilities of obstacle avoidance. Due to the computational cost of calculating avoidance paths, the maximum speed supported for obstacle avoidance is approximately 3 m/s. Other aspects to consider are the flight range, the battery life, the payload capacity, and regulatory restrictions when deploying the UAV. Adhering to these considerations ensures the UAV operates within its specified operational limits and maximizes its potential in supporting fire suppression efforts. It is important to note that the current limitation can not be corrected as it is a result of the simulated execution environment.

4.2. Unmanned Ground Vehicle

The UGV's behavior has been extensively tested and validated through simulations conducted in Gazebo, a popular robot simulation environment. The simulations were performed on a machine with an Intel Core i7-8750H 2.20 GHz CPU with 6 cores and Intel UHD Graphics, running Ubuntu 20.04 LTS with ROS version Noetic. These specifications ensured sufficient computational power and compatibility with the ROS framework for accurate simulation results. In the simulations, we created a realistic forest environment, incorporating vegetation and fire representation. The fire spots were represented by a cube, while trees acted as obstacles that the UGV had to navigate around. To evaluate the UGV's performance, we conducted a series of tests with varying distances to the fire spot. In each test, the UGV receives the geographical coordinates. Upon receiving the GPS coordinates of the fire from the UAV, it initiates its simulation and autonomously navigates through the forest (Figure 14). It utilizes the YOLOv4-Tiny model, installed on the onboard camera, to detect the fire (Figure 15). Once identified, the UGV proceeds to extinguish the fire by releasing the extinguisher. Finally, it returns to the home base.

1. In the first test, GPS coordinates were provided to the UGV, leading it to move 100 m within the environment before reaching the fire. The UGV successfully reached the fire in 5 min and 40 s.
2. In the second test, GPS coordinates were given to the UGV, instructing it to move 200 m within the environment before reaching the fire. The UGV reached the fire in 12 min and 27 s.
3. In the third test, GPS coordinates were provided to the UGV, requiring it to move 350 m within the environment before reaching the fire. The UGV reached the fire in 22 min and 12 s.

Throughout all the tests, the UGV demonstrated its ability to autonomously locate and suppress fires, successfully overcoming the obstacles posed by the simulated trees. However, it should be noted that the precision of navigation decreased as the distance to the fire spot increased. This imprecision was primarily attributed to the limitations in odometry accuracy, which was affected by the Extended Kalman Filter parameters' tuning, which is responsible for fusing data from wheel odometry, GPS, and IMU to improve odometry and enable more precise localization of the robot in space, thus enhancing navigation. It is important to note that the current imprecision can not be corrected as it is a result of the simulated execution environment. Additionally, the UGV's design is optimized for operating in extreme terrains, as indicated by the technical specifications of the simulated Jackal robot from Clearpath Robotics (<https://clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/>, accessed on 18 April 2023). However, it is important to note that the UGV's capabilities are constrained by the limitations of the simulated environment. While the robot's control system is optimized for various scenarios, it may not be able to account for every possible vegetation type or forest condition. The simulated forest environment provides a representative setting for testing and evaluation, but real-world forests can present a wide range of complexities and variations. To account for these limitations, our system incorporates an Unmanned Aerial Vehicle (UAV) as a complementary asset to assist in fire suppression operations. The UAV acts as an aerial support platform, providing additional situational awareness and intervention capabilities when the conditions for the UGV are not favorable or when inaccessible terrains are encountered. By combining the capabilities of both the UGV and UAV, our system aims to address a broader range of fire scenarios in diverse forest environments. It is important to recognize that not all terrain and forest configurations can be covered by the robots alone. In practical applications, the deployment of the UGV and UAV would need to consider access to physically reachable areas and ensure that the robots operate within their specified operational limits. While our system offers a significant advancement in autonomous fire suppression, it is crucial to align the capabilities of the robots with the characteristics and constraints of the target forest environments.

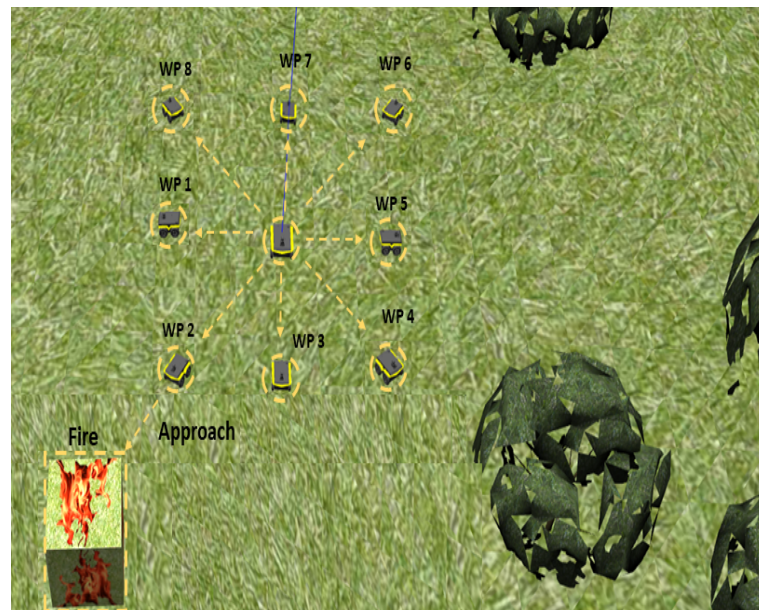


Figure 14. Navigation path pattern to detect fire.

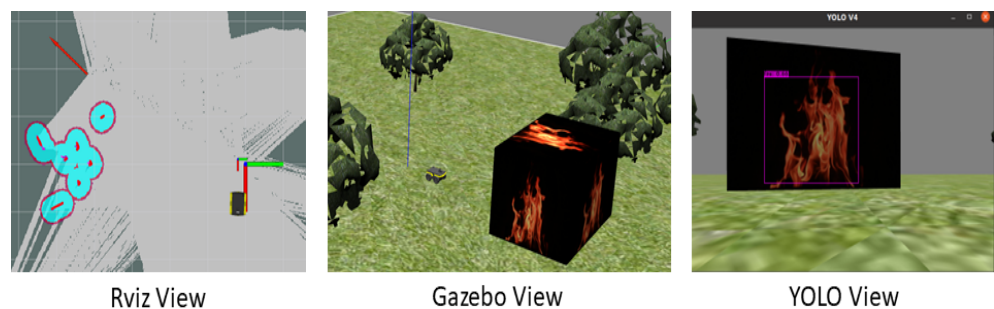


Figure 15. Search of the fire.

5. Discussion

The increasing frequency and severity of wildfires pose a significant threat to the environment, property, and human health. Early detection and rapid response are crucial to mitigating their impact and minimizing the potential devastation. This article has proposed a novel CPS architecture that integrates advanced technologies such as IoT, robotics, and UAV to provide an innovative strategy for the early detection and extinguishing of forest fires.

By utilizing low-cost IoT sensors, the proposed system is capable of continuously monitoring potential fire hazards, and taking appropriate measures to prevent a serious threat. The integration of smart cameras and infrared sensors enables real-time detection of flames, which is essential for quickly identifying and responding to wildfires. Furthermore, the system incorporates a CO₂ sensor for real-time monitoring of carbon monoxide levels in the air, which can help measure the severity of the fire and the level of danger it poses to nearby communities.

The data collected by the IoT system is sent to a ground control station, which can visualize the data through a web-based dashboard, allowing for efficient and effective decision-making in response to a wildfire. The system can also trigger appropriate response measures, such as sending alerts to local authorities and emergency services, as well as unmanned systems, for an immediate and coordinated response.

The proposed CPS architecture integrates unmanned systems, including Unmanned Aerial and Ground Vehicle, which provide the advantage of quickly identifying the exact fire location and suppressing the fires. The UAV is capable of navigating and exploring its surroundings while avoiding obstacles, detecting a fire, and releasing a fire extinguisher

ball to simulate fire suppression. Furthermore, the UAV can share geographical coordinates with the UGV to guide it to the exact fire location. The UGV can automatically navigate into the forest, reaching the fire spots, and suppressing the fire using water or a chemical extinguisher. Its control system utilizes advanced planning algorithms, including Dijkstra's algorithm for global planning and the Dynamic Window Approach (DWA) for local planning, enabling obstacle avoidance and efficient path selection.

5.1. Considerations from the Evaluation Study in a Simulated Environment

The proposed CPS architecture showcased the integration of various open-source technologies in the evaluation study conducted in a simulated ROS environment. This practical demonstration makes the architecture applicable in diverse settings, such as industries, surveillance, and precision agriculture, where cyber-physical systems can be effectively utilized. The Unmanned Aerial Vehicle (UAV) displayed its autonomous capabilities during testing, successfully locating and suppressing fires while navigating simulated tree obstacles. The UAV proved valuable in fire detection and suppression operations as an aerial support platform. It enhances situational awareness by providing a bird's-eye view of the fire and can intervene in scenarios where the Unmanned Ground Vehicle (UGV) faces limitations or challenges. By combining the UAV with the UGV and wildfire detection systems, the solution covers a broader range of fire scenarios in diverse forest environments, from rural areas to urban parks. The synergy between the UGV and UAV optimizes the efficiency and effectiveness of fire suppression operations, leveraging the unique strengths of each platform. However, it is important to consider limitations and capabilities related to obstacle avoidance, such as the computational complexity of calculating avoidance paths and the maximum supported speed. Other factors such as flight range, battery life, payload capacity, and compliance with regulations during UAV deployment require careful attention to optimize its effectiveness in supporting fire suppression efforts.

5.2. Limitations in Real-World Scenarios

The implementation of a cyber-physical system for wildfire detection and firefighting is undoubtedly a promising approach to combating forest fires. However, several real-world limitations need to be considered to ensure the effectiveness and feasibility of such a system in concrete scenarios. One significant limitation is the challenging atmospheric stability conditions present in close proximity to forest fires. The unstable nature of the environment, characterized by intense gusts of wind and sudden changes in direction, poses a considerable challenge for flying objects. Moreover, such objects would be exposed to extremely high temperatures. Stabilization at higher altitudes, while a potential solution, relies on staying within radio coverage. Unfortunately, due to line-of-sight limitations, obstructions, and the use of Wi-Fi, the effective range for maintaining reliable communication is often short. This restricted range can impede the timely transmission of critical data, potentially hindering the ability to transmit precise GPS locations of fires identified through AI analysis.

Moreover, the irregularities of the terrain in forestry areas pose another significant obstacle to the implementation of Unmanned Ground Vehicles (UGVs) for firefighting purposes. The complex topography often found in such regions makes it challenging for machines to access certain locations, depending on the kind and quality of the vehicle. Then, this limitation may prevent the system from fully capitalizing on the advantages of UGVs in navigating and operating in rugged environments. Finally, the architecture uses sensors that measure humidity, temperature, and CO₂ levels to detect fire incidents. For this, the system requires internet connectivity that can be unstable in rural areas due to poor coverage in forested areas. Addressing these limitations will be crucial to ensure the practicality and success of such a system in real-world firefighting scenarios. It entails integrating sensors and robotic vehicles into the proposed architecture that are suitable for the specific operating context. For instance, challenging environments such as rugged terrains in forested areas require vehicles with advanced or extreme capabilities; however,

these vehicles may still need to be able to access the entire area. On the other hand, more straightforward scenarios, such as patrolling large urban parks, may have fewer efficiency requirements for vehicles and sensors. Nevertheless, it is essential to carefully address all the above-mentioned limitations in the cyber-physical system's design for wildfire detection and firefighting.

6. Conclusions

This paper has presented the successful implementation of a collaborative robotic system for fire detection and suppression, utilizing the UGV platform and integrating IoT and UAV technologies. The system was validated in a simulation environment and demonstrated significant potential for improving the effectiveness and efficiency of firefighting operations by leveraging the strengths of various robotic components and enabling real-time communication and coordination. The integration of the UGV, IoT, and UAV technologies provided better situational awareness and decision-making capabilities in firefighting scenarios. The UGV, equipped with external localization and autonomous navigation, autonomously navigated through rough terrain and adverse weather conditions, identified the fire's location, and built a map to gather valuable information on the fire and surrounding environment. The IoT technology enabled real-time data monitoring for better decision-making by firefighters and provided immediate alerts to the UGV on the start of the fire and its approximate location for immediate action. The UAV technology allowed for aerial reconnaissance and data collection, enhancing the system's ability to identify fires and providing more precise fire location information to the UGV. The real-time communication and coordination capabilities of the system facilitated efficient collaboration between the robotic components and IoT, enabling quick and accurate responses to changing conditions in firefighting scenarios. Overall, the successful implementation of this collaborative robotic system for fire detection and extinguishing represents a significant contribution to the field of robotics and firefighting.

7. Future Research

Although the collaborative robotic system presented in this paper has shown promising results, there are several challenges and opportunities for future research and development:

- **Alternative Communication Technologies:** Exploring alternative communication technologies such as LoRa could extend the range and battery life of the IoT node, enhancing its capabilities in remote and energy-constrained environments.
- **Cloud Service Integration:** Investigating the integration of cloud services to store humidity and temperature data in a Cloud NoSQL database could enable the training of machine learning models for fire prediction, enhancing the proactive capabilities of the IoT system.
- **Advanced Sensor Technology:** Upgrading the IoT system's sensor technology with advanced sensors such as thermal cameras could improve its fire detection capabilities and enhance the accuracy of fire location identification.
- **Migration to ROS2:** Exploring the migration of the Rover's ROS services from ROS to ROS2 could improve the performance of the ROS node execution and ensure compatibility with the latest advancements in the ROS ecosystem.
- **Localization Optimization:** Fine-tuning the parameters of the Extended Kalman Filter (EKF) for localization could lead to further improvements in the UGV's movement precision, resulting in more efficient and effective fire extinguishing operations.
- **Swarm Robotics:** Extending the system's capabilities by implementing a swarm of drones, rovers, and IoT nodes that collaborate and communicate with each other could enhance fire detection and extinguishing capabilities, enabling more efficient and comprehensive firefighting operations.
- **Real-World Deployment:** Testing and deploying the system in real-world scenarios to evaluate its effectiveness and performance under practical conditions would provide valuable insights and validation of its capabilities.

In conclusion, our work introduces a CPS architecture specifically designed for early-stage wildfire detection and firefighting. We have proposed a state-of-the-art solution that leverages low-cost IoT sensors and ROS-based collaborative unmanned robots to address the critical issue of wildfires. It is crucial to emphasize that our research is currently in an initial phase. The system we have developed is primarily intended to detect and combat wildfires at their early stages, rather than when they have already escalated. We acknowledge that the effectiveness of our system has been analyzed solely in simulated scenarios, and there are still ongoing development challenges that may arise in real-world situations. Therefore, further refinement and real-world experimentation are necessary to validate the performance and effectiveness of our proposed architecture. We are aware of the need for additional work to address potential challenges that may be encountered in actual fire scenarios. The future research directions outlined above offer exciting possibilities for further advancements in this field, ultimately leading to more advanced and capable systems in the future. This work contributes valuable insights and sets the stage for continued research and development in the area of collaborative robotic systems for firefighting.

Author Contributions: P.B., A.A.C., G.M., V.P., M.R., M.S. and G.V. contributed to the design and implementation of the research, to the analysis of the results and to the writing of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ginige, A.; Paolino, L.; Romano, M.; Sebillio, M.; Tortora, G.; Vitiello, G. Information Sharing Among Disaster Responders—An Interactive Spreadsheet-Based Collaboration Approach. *Comput. Support. Coop. Work* **2014**, *23*, 547–583. [CrossRef]
2. Battistoni, P.; Romano, M.; Sebillio, M.; Vitiello, G. Monitoring Urban Happiness through Interactive Chorems. *Sustainability* **2023**, *15*, 8733. [CrossRef]
3. Romano, M. Urban Happiness Planning Through Interactive Chorems. In Proceedings of the International Symposium: New Metropolitan Perspectives, Reggio Calabria, Italy, 24–26 May 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 2822–2832.
4. Battistoni, P.; Grimaldi, M.; Romano, M.; Sebillio, M.; Vitiello, G. Interactive maps of chorems explaining urban contexts to align smart community's actors. In Proceedings of the International Conference on Computational Science and Its Applications, Cagliari, Italy, 13–16 September 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 549–564.
5. Romano, M.; Díaz, P.; Aedo, I. Gamification-less: May gamification really foster civic participation? A controlled field experiment. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 4451–4465. [CrossRef]
6. Romano, M.; Díaz, P.; Aedo, I. Emergency management and smart cities: Civic engagement through gamification. In Proceedings of the Information Systems for Crisis Response and Management in Mediterranean Countries: Third International Conference, ISCRAM-med 2016, Madrid, Spain, 26–28 October 2016; Proceedings 3; Springer: Berlin/Heidelberg, Germany, 2016; pp. 3–14.
7. Wang, H.; Liu, J.; Han, J. RS-CPS: A distributed architecture of robotic surveillance cyber-physical system in the nature environment. In Proceedings of the 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Shenyang, China, 8–12 June 2015. [CrossRef]
8. Singh, K.K.; Nayyar, A.; Tanwar, S.; Abouhawwash, M. *Emergence of Cyber Physical System and IoT in Smart Automation and Robotics-Computer Engineering in Automation*; Advances in Science, Technology & Innovation; SpringerLink: Berlin/Heidelberg, Germany, 2021.
9. Goyal, N.; Sharma, S.; Rana, A.K.; Tripathi, S.L. *Internet of Things Robotic and Drone Technology*; CRC Press: Boca Raton, FL, USA; Taylor & Francis Group: Abingdon, UK, 2022.
10. Gregorio, M.D.; Romano, M.; Sebillio, M.; Vitiello, G.; Vozella, A. Improving Human Ground Control Performance in Unmanned Aerial Systems. *Future Internet* **2021**, *13*, 188. [CrossRef]
11. Gandhi, A.; Garg, S.; Parkhiya, P.; Zhu, Z. UAV - UGV Collaborative Robots for Fire-Fighting. Available online: https://mrsdprojects.ri.cmu.edu/2018teamh/wp-content/uploads/sites/38/2019/05/TeamH_CDRR.pdf (accessed on 1 May 2023).

12. Zhao, L.; Zhi, L.; Zhao, C.; Zheng, W. Fire-YOLO: A Small Target Object Detection Method for Fire Inspection. *Sustainability* **2022**, *14*, 4930. [CrossRef]
13. Zadeh, N.R.N.; Abdulwakil, A.H.; Amar, M.J.R.; Durante, B.; Santos, C.V.N.R. Fire-fighting UAV with shooting mechanism of fire extinguishing ball for smart city. *Indones. J. Electr. Eng. Comput. Sci.* **2021**, *22*, 1320–1326. [CrossRef]
14. Ghamry, K.A.; Kamel, M.A.; Zhang, Y. Cooperative forest monitoring and fire detection using a team of UAVs-UGVs. In Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1206–1211.
15. Ray, P.P. Internet of robotic things: Concept, technologies, and challenges. *IEEE Access* **2016**, *4*, 9489–9500. [CrossRef]
16. Mathiassen, K.; Baksas, M.; Græe, S.A.; Mentzoni, E.A.; Nielsen, N.H. Making the Milrem Themis UGV ready for autonomous operations. In Proceedings of the Unmanned Systems Technology XXIII, Online, 12–17 April 2021. [CrossRef]
17. San-Miguel-Ayanz, J.; Durrant, T.; Boca, R.; Maianti, P.; Liberta', G.; Artes Vivancos, T.; Jacome Felix Oom, D.; Branco, A.; De Rigo, D.; Ferrari, D.; et al. Forest Fires in Europe, Middle East and North Africa 2021. In *JRC Technical Report—European Commission*; Publications Office of the European Union: Luxembourg, 2021. [CrossRef]
18. Grant, C.; Hamins, A.; Bryner, N.; Jones, A.; Koepke, G. *Research Roadmap for Smart Fire Fighting*; National Institute of Standards and Technology: Gaithersburg, MA, USA, 2015. [CrossRef]
19. Ai, W.; Wu, Y.; Wang, C.; Liang, L. Impacts of Climate Change on Forest Fires. In *Annual Report on Actions to Address Climate Change (2019)*; Springer: Singapore, 2023; pp. 195–209. [CrossRef]
20. Barmpoutis, P.; Papaioannou, P.; Dimitropoulos, K.; Grammalidis, N. A Review on Early Forest Fire Detection Systems Using Optical Remote Sensing. *Sensors* **2020**, *20*, 6442. [CrossRef] [PubMed]
21. Kresek, R. History of the Osborne Firefinder. Available online: <http://www.nysforestrangers.com/archives/osborne%20firefinder%20by%20kresek.pdf> (accessed on 27 May 2023).
22. Mohapatra, A.; Trinh, T. Early Wildfire Detection Technologies in Practice—A Review. *Sustainability* **2022**, *14*, 12270. [CrossRef]
23. Sanislav, T.; Mois, G.; Folea, S.; Miclea, L.; Gambardella, G.; Prinetto, P. A cloud-based Cyber-Physical System for environmental monitoring. In Proceedings of the Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 15–19 June 2014; pp. 6–9. [CrossRef]
24. Sahal, R.; Alsamhi, S.H.; Breslin, J.G.; Ali, M.I. Industry 4.0 towards Forestry 4.0: Fire detection use case. *Sensors* **2021**, *21*, 694. [CrossRef] [PubMed]
25. Partheepan, S.; Sanati, F.; Hassan, J. Autonomous Unmanned Aerial Vehicles in Bushfire Management: Challenges and Opportunities. *Drones* **2023**, *7*, 47. [CrossRef]
26. Esfahlani, S.; Cirstea, S.; Sanaei, A.; Cirstea, M. Fire detection of Unmanned Aerial Vehicle in a Mixed Reality-based System. In Proceedings of the IECON 2018—44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018; pp. 2757–2762. [CrossRef]
27. Jayapandian, N. Cloud Enabled Smart Firefighting Drone Using Internet of Things. In Proceedings of the International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 27–29 November 2019; pp. 1079–1083. [CrossRef]
28. Xu, R.; Lin, H.; Lu, K.; Cao, L.; Liu, Y. A forest fire detection system based on ensemble learning. *Forests* **2021**, *12*, 217. [CrossRef]
29. Nori, R.; Farhan, R.; Abed, S. Indoor and Outdoor Fire Localization Using YOLO Algorithm. *J. Phys. Conf. Ser.* **2021**, *2114*, 012067. [CrossRef]
30. Bahhar, C.; Ksibi, A.; Ayadi, M.; Jamjoom, M.M.; Ullah, Z.; Soufiene, B.O.; Sakli, H. Wildfire and Smoke Detection Using Staged YOLO Model and Ensemble CNN. *Electronics* **2023**, *12*, 228. [CrossRef]
31. Fonollosa, J.; Solórzano, A.; Marco, S. Chemical Sensor Systems and Associated Algorithms for Fire Detection: A Review. *Sensors* **2018**, *18*, 553. [CrossRef] [PubMed]
32. Gaur, A.; Singh, A.; Kumar, A.; Kulkarni, K.; Lala, S.; Kapoor, K.; Srivastava, V.; Kumar, A.; Mukhopadhyay, S. Fire Sensing Technologies: A Review. *IEEE Sens. J.* **2019**, *19*, 3191–3202. [CrossRef]
33. Liu, P.; Yu, H.; Cang, S.; Vladareanu, L. Robot-assisted smart firefighting and interdisciplinary perspectives. In Proceedings of the 22nd International Conference on Automation and Computing (ICAC), Colchester, UK, 7–8 September 2016; pp. 395–401. [CrossRef]
34. Benzekri, W.; Moussati, A.E.; Moussaoui, O.; Berrajaa, M. Early Forest Fire Detection System using Wireless Sensor Network and Deep Learning. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*. [CrossRef]
35. Avazov, K.; Hyun, A.E.; Sami, A.A.; Khaitov, A.; Abdusalomov, A.B.; Cho, Y.I. Forest Fire Detection and Notification Method Based on AI and IoT Approaches. *Future Internet* **2023**, *15*, 61. [CrossRef]
36. Bushnaq, O.M.; Chaaban, A.; Al-Naffouri, T.Y. The role of UAV-IoT networks in future wildfire detection. *IEEE Internet Things J.* **2021**, *8*, 16984–16999. [CrossRef]
37. Di Gregorio, M.; Nota, G.; Romano, M.; Sebillio, M.; Vitiello, G. Designing Usable Interfaces for the Industry 4.0. In Proceedings of the International Conference on Advanced Visual Interfaces, Salerno, Italy, 28 September–2 October 2020. [CrossRef]
38. Battistoni, P.; Sebillio, M.; Vitiello, G. An IoT-Based Mobile System for Safety Monitoring of Lone Workers. *IoT* **2021**, *2*, 476–497. [CrossRef]

39. Cameron Hughes, T.H. *Robot Programming: A Guide to Controlling Autonomous Robots*; Que: Upper Saddle River, NJ, USA, 2016; p. 309.
40. Avolicino, S.; Di Gregorio, M.; Romano, M.; Vitiello, G.; Santoro, E.; Landolfi, E. C-MOBILITY: Leveraging the Drivers' Situation Awareness through a Mobile Application. In *Proceedings of the 2022 International Conference on Advanced Visual Interfaces*, Rome, Italy, 6–10 June 2022; Association for Computing Machinery: New York, NY, USA, 2022. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.