FireVision: An Early Fire and Smoke Detection Platform Utilizing Mask-RCNN Deep Learning Inferences

Konstantina Spanoudaki , Meropi Tsoumani , Sotirios Kontogiannis * , Myrto Konstantinidou , Ion Anastasios Karolos , George Kokkonis

Posted Date: 20 January 2026

doi: 10.20944/preprints202601.1534.v1

Keywords: fire detection system; drone flight algorithm; deep learning; ensemble learning; object detection; instance segmentation; edge computing

 doi:10.20944/preprints202601.1534.v1

Article

FireVision: An Early Fire and Smoke Detection Platform Utilizing Mask-RCNN Deep Learning Inferences

Konstantina Spanoudaki [1], Meropi Tsoumani [1], Sotirios Kontogiannis [1,*], Myrto Konstantinidou [2], Ion Anastasios Karolos[3] and George Kokkonis[4]

[1] Laboratory Team of Distributed Microcomputer Systems, Department of Mathematics, University of Ioannina, 45110 Ioannina, Greece
[2] Systems Reliability and Industrial Safety Laboratory, Institute for Nuclear and Radiological Sciences, Energy, Technology and Safety, NCSR Demokritos, Ag. Paraskevi, 15341 Athens, Greece
[3] School of Rural and Surveying Engineering, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece
[4] Department of Information and Electronic Engineering, International Hellenic University, 57001 Thessaloniki, Greece)
* Correspondence: skontog@uoi.gr

Abstract

This paper introduces FireVision, a novel detection platform and model designed for real-time fire detection and monitoring. The platform employs automatic drone flights to capture high-resolution images across both suburban and forest environments. It uses ensemble deep learning inference from Mask R-CNN weak learners to trigger alerts. These inferences are enhanced by the accurate detection capabilities of ResNet-50, ResNet-101, and ResNet-152 classifiers, which can be deployed either in the cloud or on the drone's edge co-processing units. The authors also implemented an index, called the Fire Criticality Index, that uses detection bounds and masks to indicate the criticality of a fire event, as well as an automated drone path-planning algorithm for detecting a fire critical event. The authors conducted experiments with their proposed model using a mask-annotated dataset comprising of 12,000 images. They evaluated the model's accuracy and inference speed across various cloud and edge computing setups. The experimental results revealed that ResNet-101 outperformed ResNet-50 by 5–12.5% in maP@0.5 mask accuracy and demonstrated an 18% increase in inference time when executed on the cloud and a 27% increase on the drone edge device GPU. For the ResNet-152 compared to ResNet-101, the map@0.5 increased by 0.5–1.2%. However, the ResNet-152 inference time was 9x slower in the cloud and 1.3x slower on the GPU.

Keywords: fire detection system; drone flight algorithm; deep learning; ensemble learning; object detection; instance segmentation; edge computing

1. Introduction

Wildfires are one of the most significant challenges of the modern world, affecting various sectors, including the environment, economy, and public health [1,2]. Their occurrence has a profound impact on human health, as a variety of harmful substances are released into the atmosphere, including particulate matter ($PM_{2.5}$), toxic pollutants, and volatile organic compounds, all of which have been linked to increased human sickness and mortality [3–5]. Additionally, wildfires devastate the environment, natural landscapes, resources, and infrastructure, resulting in substantial environmental and economic losses, as well as long-term effects on ecosystems and agricultural production [6]. In many instances, wildfires endanger buildings and entire communities, often leading to tragic loss of life, especially in wildland–urban interface areas [7,8]. The human factor significantly contributes to the escalation of this issue, as a considerable percentage of wildfires are caused by human negligence or improper practices [9]. Many of these disasters result from deficiencies in prevention efforts and misplanning,

 doi:10.20944/preprints202601.1534.v1

as well as technological challenges in continuously monitoring large areas. This pinpoints the urgent need for early detection and effective wildfire management strategies [10–13].

Numerous efforts have been made to address wildfire detection; however, recent studies confirm the need to transition from traditional sensor-based systems to approaches based on deep learning. These methods offer faster and more accurate detection, the ability to monitor and cover larger areas, and a reduced number of false alarms, which are often triggered in sensor-based systems by environmental factors such as humidity, dust, and exhaust gases [14,15]. Moreover, deep learning systems based on image or video data can operate effectively in real time, making them particularly suitable for large-scale early wildfire detection applications [10].

Approaches to wildfire detection using deep learning include image analysis and techniques such as image classification, object detection, and instance segmentation (pixel masking). Among the most widely used approaches are object detection [16] models, such as YOLO [17], as well as instance segmentation methods [18], such as Mask R-CNN [19]. Recent studies comparing the accuracy and speed of these models indicate that, although YOLO-based models outperform in terms of inference speed and are well-suited for real-time applications, instance segmentation methods such as Mask R-CNN achieve higher accuracy in detecting small-scale or structurally complex objects, providing more precise localization of fire regions [20–22]. Implementations of Mask R-CNN provided by the torchvision library [23] (ResNet-50 implementation only), community-contributed repositories of implementations [24], and high-performance instance segmentation frameworks like Detectron2 by Facebook AI [25] are open-source, accessible, and extensible tools for deploying these models in wildfire detection systems.

For this reason, and given that the present study focuses on early fire and smoke detection, the Mask R-CNN model was selected, as it enables accurate localization and detailed pixel-level segmentation of regions of interest. Specifically, this study compares three variants of Mask R-CNN that use Residual block classifiers. That is, employing ResNet-50, ResNet-101, and ResNet-152 [26,27] as backbone networks for their proposed fire detection model, which differ in terms of architectural complexity. The evaluation focuses on both detection accuracy and model inference time.

Regarding the monitoring of smoke and fire incidents using dense sensor grids, a set of requirements must be met that include connectivity (ZigBee, Wi-Fi, NBIoT or LoRaWAN), sensing devices autonomous operation (battery operated and use of photovoltaic panels for charging even in dim light), accessibility to the installation location, maintenance, and cost (reliable low cost sensors of minimum calibration efforts). Depending on the placement density of autonomous sensing devices, a forest fire can be classified according to [11] to classes A–F, and by using LSTM and GRU models [28], the fire event can be either detected or its evolution can be locally predicted. Smoke incidents can be detected utilizing sensors that measure particulate matter ($PM_{2.5}/PM_{10}$), humidity, infrared, barometric pressure, and gases (CO, CO_2, Volatile Organic Compounds - VOCs). In addition, for fire detection, temperature, heat, gas, flame, and Go-based sensors are used [29]. These sensors are low-cost, low-power, and suited for dense deployments, enabling continuous, near-real-time operations. For urban installations, sensory maintenance and proximity are feasible; however, in non-urban environments, particularly in forest environments, sensors' autonomous operation, proximity, and maintenance are challenging and require significant human effort for drone-based sensing.

The effectiveness of early smoke or fire detection systems is also strongly influenced by sensor deployment strategies. It has been emphasized in [30,31] that optimal Wireless Sensor Network (WSN) design is crucial for protecting forest areas and critical infrastructure, and simulations have demonstrated that optimized sensor placement can significantly reduce response time before a fire threatens critical infrastructure and can also complement vision-based detection systems. For industrial, infrastructures, and urban environments, false alarms caused by dust or vapor remain a significant challenge. To address this issue, a visual sensor that integrates hybrid deep learning sensing techniques was developed in [32]. This approach focuses on filtering environmental disturbances that visually resemble smoke, providing a reliable solution for safeguarding critical infrastructures.

 doi:10.20944/preprints202601.1534.v1

Furthermore, the non-necessity of autonomous sensors in urban and industrial environments makes sensor deployments a key implementation candidate, with camera-based vision and object detection as complementary monitoring tools. Specifically, for ATEX (ATmosphères EXplosibles) environments and military installations, where the use of drones is strictly prohibited. Nevertheless, this is not the case in forest and non-urban environments, where most catastrophic, uncontrolled fire incidents occur. This paper focuses solely on this kind of environment for deploying the authors' proposed architecture.

Focusing on remote sensing, NASA's Fire Information for Resource Management System (FIRMS) system provides periodic fire detections derived from satellite imagery [33]. FIRMS primarily uses data from the MODIS sensors on the Terra and Aqua satellites and the VIIRS sensors on the Suomi NPP and NOAA-20 satellites. These instruments acquire thermal infrared imagery multiple times per day, enabling the detection of active fires based on thermal anomalies at spatial resolutions of approximately 1 km (MODIS) and 375 m (VIIRS), as well as burn area products [34]. Similarly, the European Forest Fire Information System (EFFIS) [35] uses the EU Sentinel-2,3 satellites for active fire detection by providing high-resolution optical imagery (10–20 m spatial resolution) through MultiSpectral Instruments (MSI). Although Sentinel satellite instruments are not optimized for thermal anomaly detection, their fine spatial resolution multirpectral capabilities are highly valuable for post-fire assessment, burned area mapping, and identification of fire scars and vegetation damage [36,37].

Satellite-based fire detection methods using multispectral and thermal sensors have also been explored, enabling large-scale monitoring [38,39]. However, they suffer from limited spatial resolution and sensitivity to environmental factors such as cloud coverage, often requiring substantial fire spread before reliable detection is possible [40]. Furthermore, they operate periodically rather than in close to real time for the acquisition of sensory imagery data, outside the US [41]. Overall, robust vision-based fire segmentation at the pixel level remains an open and up-and-coming research area with significant potential for further advancement, particularly with the new hyperspectral image capabilities of new satellite launches [42], particularly in the EU [43].

Regarding automatic drone flights and edge intelligence, drones, specifically on the DJI Matrice platform, are achieved through the integration of onboard and payload-level computing, the DJI Manifold 3 embedded AI computer [44]. The Manifold 3 is based on the NVIDIA Jetson Orin NX architecture, for real-time execution of deep learning models directly on the drones without reliance on cloud infrastructure. Therefore, allowing object detection and instance segmentation models for fire and smoke at the drone level with latencies on the order of milliseconds for object detection (specifically, YOLO models). In terms of flight coverage, the DJI Matrice 4TD operating within an automated dock-based deployment can achieve significant area surveillance both per flight and cumulatively, achieving a maximum flight time of approximately 54 minutes and an operational radius of up to 10 km. At altitudes around 100–120m, an area of approximately 1.2 km^2 per frame can be covered, enabling rapid area scanning, while the zoom and thermal sensors allow targeted inspection of fire-detected anomalies [45]. When combined with automated recharging and alternating Dock stations, near-continuous monitoring of large forested regions becomes feasible, effectively bridging the gap between local high-resolution sensing and persistent surveillance.

In this paper, the authors propose a low-cost architecture that uses cameras and automatically planned drone flights to monitor non-urban locations and forests for early detection of fire incidents. Their architecture is supported by a novel model called FireVision. The authors' proposed model incorporates a two-layer Mask R-CNN approach: first for smoke detection, then for fire detection within the bounding boxes of masked contours. A fire-expansion classification algorithm enhances the dual-layer approach called the Fire criticality algorithm, which aims to quantify both the affected area and the fire intensity through a fire-to-smoke index, the fire criticality index. The comparison of inference time of the proposed FireVision model is conducted across two distinct execution environments: 1) an edge computing device (NVIDIA Jetson nano [46]) with limited computational resources, included as an extension board corposessor on most commercial drones [47,48], and 2) a cloud computing Virtual

 doi:10.20944/preprints202601.1534.v1

Privacy Service (VPS) systems with GPU processing capabilities. Therefore, it allows assessing the suitability of each inference environment across different deployment scenarios; edge and cloud-based inferences. This paper's comparative analysis also examines the impact of distinct backbone selections on inference performance and the practical applicability of Mask R-CNN for early wildfire detection in real-world conditions.

The remainder of the paper is organized as follows: Section 2 focuses on related work in the field of fire and smoke detection; Section 3 describes the materials and methods used in this study and the authors' model proposition and algorithmic process for fire detection called FireVision. Section 4 presents the FireVision experimental scenarios and reports the experimental results along with their discussion; and finally, Section 5 concludes the paper.

2. Related Work

In recent years, due to climate change and advances in remote sensing, the need for early fire detection has become increasingly critical. As a result, fire detection has attracted substantial research interest, particularly in the context of deep learning–based visual analysis. In this section, existing approaches are categorized into three main detection classes. These are fire and smoke detection approaches, as well as bilateral approaches. Existing methods are organized by detection target and the underlying technologies employed. A comparative discussion highlights the capabilities and limitations of current approaches relative to the proposed FireVision detection process.

2.1. Smoke Detection

Since smoke typically appears before visible flames, smoke detection has become a critical component of early fire detection. For this reason, recent research has increasingly focused on developing smoke detection methods, including image detection of either deep learning techniques or wireless sensing systems using infrared, CO, CO_2, and air particle concentration sensors.

Using RGB cameras and deep learning object detection models, one of the main challenges arises from the air-sustained diffusion and the visually transparent nature of smoke. Early approaches, such as the DeepSmoke model [49] and the target-aware method [50], focused on distinguishing smoke-related features from static background information. However, deep learning models' ability to learn from new data without losing previously acquired knowledge remains challenging. To address this limitation, transfer learning strategies combined with Learning without Forgetting (LwF) mechanisms have been explored, achieving high classification accuracy of around 96.9% for smoke and fire images, while maintaining previously learned representations [51]. Reusing trained models on large datasets like ImageNet as a starting point for a new but related task (e.g., wildfire detection) allows leveraging existing knowledge to improve learning efficiency and performance, especially when labeled data is limited [52]. Furthermore, Learning without Forgetting (LwF) principles include mechanisms such as an incremental learning process that combines the standard detection loss for new classes with an appropriate loss function (distillation loss) that forces the model to produce similar responses for both old and new classes [53], similar to stranded LSTM models for sensory data inputs [54]. Additionally, selective knowledge transfer from the teacher (old model) to the student (new model) across classification and bounding-box regression outputs may further alleviate forgetting [55].

The demand for real-time incident response has further driven the development of specialized architectures and edge-computing approaches that balance detection accuracy and processing speed. A significant contribution in this area is presented in [56], where the authors utilize an optimized YOLOv5 model for fire detection in video streams. This is achieved by introducing architectural modifications, including dilated convolutions within the Spatial Pyramid Pooling (SPP) module of the YOLO model, which, in turn, significantly improve the detection of small targets, such as early fire sources or faint smoke occurrences, while maintaining high inference speeds. Moreover, extending real-time detection capabilities, a transformer-based architecture is introduced in [57]. Unlike the convolutional neural networks (CNNs), transformer model architectures excel at capturing long-range dependencies and contextual relationships in visual data. Therefore, the introduced Triplet Attention

 doi:10.20944/preprints202601.1534.v1

mechanisms refine feature representations and highlight cross-dimensional relations across spatial and channel dimensions. Additionally, the proposed architecture integrates a High Selectivity Feature Pyramid Network (HS-FPN) for feature fusion. HS-FPN improves upon standard feature pyramid networks by selectively integrating and refining features across different resolution levels, yielding richer, more discriminative representations across different image scales.

2.2. Fire Detection

Thermal imaging cameras play a critical role in forest fire detection systems by capturing infrared radiation to identify fire hotspots based on temperature, even in the presence of smoke or foliage. These cameras can detect temperature differences as small as 0.1°C, enabling early identification of fires before flames become visible. By analyzing temperature anomalies, thermal sensors can differentiate between solar-heated surfaces and combustion zones, thereby minimizing false alarms caused by reflective materials or animal heat signatures. When integrated with RGB cameras on multi-spectral drone platforms, thermal data enables algorithmic cross-verification of visible smoke and elevated temperature profiles. Furthermore, wireless sensor networks (WSNs) comprised of ground-based may utilize acoustic sensors that also can detect the distinct low-frequency rumble of fire, as well as temperature, humidity, and infrared sensors across wide forested areas.

Focusing on the use of RGB images for fire detections, preliminary approaches utilize convolutional neural networks, such as InceptionV3-type architectures, address the problem as a binary classification task (fire/smoke versus no fire/smoke), achieving satisfactory accuracy but without providing any spatial information regarding the location or extent of the fire [58]. Subsequently, research shifted toward object detection and instance segmentation approaches, achieving real-time or near-real-time performance. Regarding real-time object detection, the widespread adoption of YOLO-based models [59] as well as two-stage architectures such as RetinaNet, which uses focal loss to handle class imbalance and is effective in forest fire smoke recognition tasks [60], as well as the slower-inference but more accurate Faster R-CNN models, or their fast mobileNet alternatives [61]. These methods enable fire localization through bounding boxes and support real-time operation in both outdoor and indoor environments [62–64]. The high-level characteristics and capabilities of object detection are presented in Table 1.

Nevertheless, object detection approaches remain limited, as bounding boxes do not accurately represent the spatial distribution of fire and often include significant non-fire regions. For this reason, a more suitable representation of fire extent may be achieved through instance-level or semantic segmentation models, which provide pixel-wise localization. Despite their potential, segmentation-based approaches, specifically instance segmentation models, remain largely underexplored. A representative example is the application of Mask R-CNN for fire segmentation. However, this method is limited to fire-only scenarios, as accurately modelling smoke at the pixel level remains particularly challenging in both inference and annotation terms. Furthermore, existing approaches rely on small-scale datasets and do not report standardized evaluation metrics such as Intersection over Union (IoU) or AP_mask, making objective comparison difficult [65].

 doi:10.20944/preprints202601.1534.v1

Table 1. Comparison of object detection architectures

Model	Architecture Type	Speed (FPS)	Accuracy (mAP)	Real-Time Suitability / Notes
RetinaNet	One-stage detector with Focal Loss	Medium	High	Balances accuracy and speed; effective for class-imbalanced tasks
Faster R-CNN (MobileNet)	Two-stage with lightweight backbone (MobileNet)	Medium-High	Medium	Slower than YOLO but more accurate than other mobile detectors
Faster R-CNN (ResNet)	Two-stage	Medium-Low	Medium	Slower than YOLO but most accurate than other detectors
YOLO (v5/v8/v11)	One-stage, end-to-end detection	High	Medium-Low	Optimized for real-time; widely used in wildfire detection
SSD	One-stage with default boxes (anchors)	High	Low	Simpler and faster than RetinaNet but slightly lower accuracy
Transformers (e.g., DETR)	Attention-based (encoder-decoder)	Low	High	High accuracy and global reasoning but slower inference; not yet optimal for real-time edge deployment

2.3. Fire and Smoke Detection

Although standalone smoke or flame detection has demonstrated satisfactory performance, simultaneous detection leverages the complementary characteristics of the two phenomena. Smoke typically appears during the early stages of a fire, whereas flames are more visually salient under low-light conditions, making their combination particularly effective for early warning systems.

Recent studies have focused on comparing different versions of the YOLO algorithm, aiming to identify the optimal balance between accuracy and inference speed. An evaluation of YOLOv5 through YOLOv8, as well as YOLO-NAS, was conducted on the Foggia dataset [66]. The results indicated that YOLOv5, YOLOv7, and YOLOv8 achieve a well-balanced performance, while YOLO-NAS outperforms the others in terms of Recall, reducing missed detections at the cost of lower Precision. This finding is particularly significant for applications where minimizing false negatives is critical.

Extending the comparison to more recent architectures, recent research [67] examined YOLOv9, YOLOv10, and YOLOv11 for smoke and fire detection. The results highlighted YOLOv11n (nano version) as the most efficient solution, achieving high performance (Precision 0.845, Recall 0.801) with significantly lower computational cost, making it especially suitable for real-time applications and embedded systems. Moreover, a new lightweight detection model called FireNet was implemented in [68]. Firenet uses a single-stage detection process and a structure similar to YOLO-family detectors. It is an object detection model, not a mask model, achieving mAP@0.5 similar to YOLOv11n(nano) with a slight 1% improvement in inference speed.

To further enhance discrimination between smoke and flames, particularly when their visual characteristics (e.g., color and texture) are ambiguous or overlap, modifications to the core YOLO architectures have been proposed. The YOLOv11-DH3 model was introduced [69], in which traditional convolutions in the detection head are replaced with Deformable Convolutional Networks v3 (DCNv3). This approach improves the network's adaptability to geometric deformations of flames, leading to more accurate detection in dynamic environments.

The use of larger and more powerful YOLO models has also been investigated. It has been shown [59] that YOLOv11x (extra large) achieves outstanding performance (mAP50 of 0.901). However,

 doi:10.20944/preprints202601.1534.v1

smoke detection remains more challenging than flame detection due to the high variability of smoke textures, often necessitating deeper, more computationally intensive networks.

Another significant challenge in simultaneous smoke and fire detection is the wide variation in object scale, particularly in drone-captured images, where small fire sources may coexist with large smoke plumes. To address this issue, an enhanced version of YOLOv8 incorporating attention mechanisms was proposed [70]. By replacing full convolutions with local convolutions in the C2F module and introducing Efficient Multi-scale Attention (EMA) and BiFormer mechanisms, a substantial accuracy improvement (93.57%) was achieved in complex forest environments.

Finally, research has also extended to the very early stages of fire ignition. The YOLO-DKM model [71] aims to detect not only flames and smoke but also sparks. By integrating the SKAttention mechanism into the backbone and employing dynamic convolutions (DSConv), the model enhances the detection of extremely small targets, providing a more comprehensive and timely early warning solution.

3. Materials and Methods

In this section, the authors present the proposed system architecture, the algorithms adopted for fire and smoke detection, and the metrics used to evaluate their performance.

3.1. Proposed FireVision System

The authors propose an automated wildfire and smoke detection system based on deep learning models. The architecture is divided into three stages: data collection, data processing, and final reporting, as shown in Figure 1

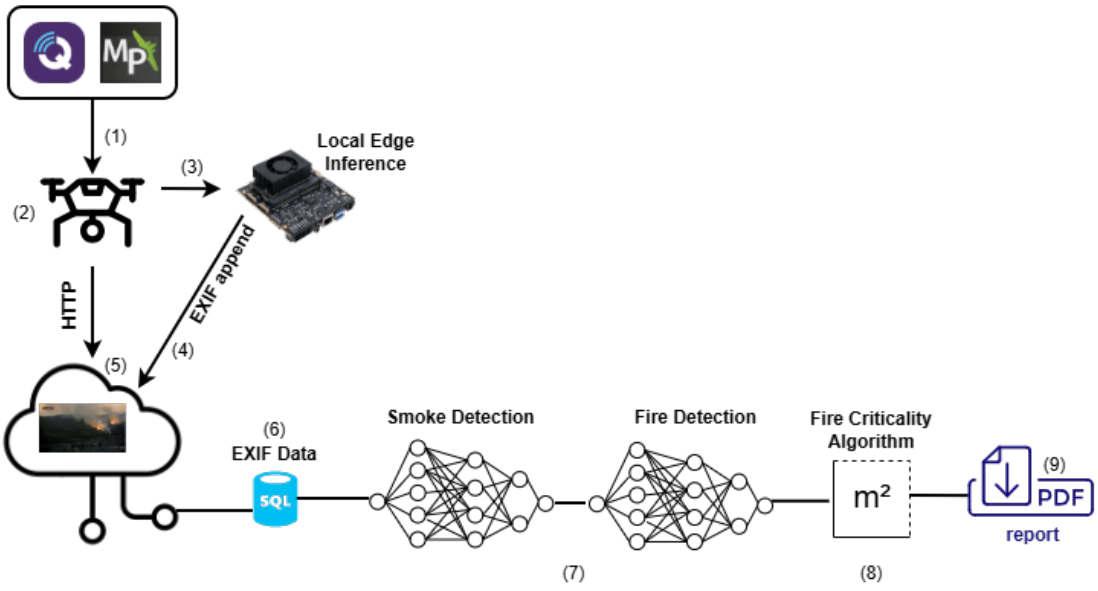

Figure 1. FireVision High Level Architecture

Initially, drone flights are organized using COTS (Commercial Off-The-Shelf) software (see Figure 1.(1)), such as Mission Planner [72] or QGroundControl [73], or PX4 autopilots [74] or the DJI bridge [75], using or following the MAVLink XML control protocol over TCP [76,77] (mavlink-router). For this enterprise setup, a static IP address is required for the LTE drone modem. In the FireVision cloud setup, this capability is provided via a private IP address using the cloud FireVision VPN service. These programs provide the capability for precise definition of geographic surveillance coordinates, ensuring the coverage of large and difficult-to-access areas during the drone flight (see Figure 1.(2)), where traditional surveillance methods fail to operate effectively, specifically in forests. The system

 doi:10.20944/preprints202601.1534.v1

provides real-time detection of fire and smoke objects, as well as instance segmentation. This capability is acquired using two alternative setup cases:

Drone edge inference. Processing is performed directly on the drone's microprocessor (e.g., Jetson Nano [46,78,79], where have previously edge potential experimented by the authors in [80]) (see Figure 1.(3)). The detection results, along with the EXIF position metadata, are then sent to the cloud (see Figure 1.(4)).

Cloud inference: Raw image data is sent in real-time directly to the FireVision cloud (see Figure 1.(5)), for object detection and instance segmentation.

In both cases (drone or cloud inferences), the detection and processing steps are the same: a) Store image meta information of GPS coordinates (EXIF), b) Execute inferences for the Mask-RCNN smoke model, c) Execute inferences for the Mask-RCNN fire model. d) From the detected contours and mask areas, calculate the Fire Criticality index using the Fire stress algorithm, and e) report back to the drones navigation process updated GPS monitoring coordinates, as instructed or not by the Fire criticality or stress algorithm.

For each drone image capture following a predifined pathplanning operation, the EXIF location data and detection information, along with the detected image, is stored in an SQL database (cloud case) or SQLite (edge case) (see Figure 1.(6)). This data storage is significantly specifically for the drone case for the extraction of flight information and its consequtive detections-images, for the process of reporting (see Figure 1.(9)).

During the image processing stage, a pre-trained detection model is used for model inference (see Figure 1.(7)), which achieves both the detection and segmentation of the fire and smoke pixels separately using two different models, pretrained on annotated data for smoke and fire, respectively. This way, an accurate definition of smoke and fire boundaries and pixel polygons (instances) is obtained, as well as two values of bounding box and pixel areas for fire and two values for smoke, and the Fire Criticality Index (FCI) is calculated (see Section 3.4). This entire process should remain in a fraction of 1–2 seconds.

Following detection, the FCI index calculation according to section 3.4, the fire criticality-stress index algorithm is executed, updating the drone's pathplanning course (see Figure 1.(8)). Based on this algorithm, the final stage involves a set of alternate points of hierarchical routing that the drone must immediately follow, added in a FIFO queue (Drone Flight Deviation Queue -DFDQ), either directly for the edge computation or as an HTTP response to an image http request. If a new pathplaning update is required to be provided to the drone the "update-pathplan" key value is also encoded as EXIF metadata dictionary of the captured drone image as a binary value 0 or 1, if the drone detects FCI value or mean FCI value of a polygon-route above threshold as instructed by the end of a revolution of the Fire criticality-stress algorithm or the end of a closed circular loop waiting for the cloud based mean FCI calculation.

The drone's MAVLink positioning and control channel uses 4G/5G via the MAVLink router messaging service, which is instantiated over TCP port 5760. The infrastructure, coverage, and system are mainly managed and monitored by the flight control software. The DFDQ's commands for drone path plan deviations are automatically appended to the original flight plan queue as a priority execution plan. Then the original flight resumes. If, however, the drone's battery is below 12%, the return-to-HOME process initiates, altering the drone's path plan and forcing it to return to base.

Finally, the FireVision system dashboard is capable of exporting a detailed report in PDF format (see Figure 1.(9)), for easier understanding and official documentation. This can be performed either in close to real-time for cloud inferences, post-processing for inferences, or after the drone acquires the relevant data. The following section describes the FireVision detection model in detail.

3.2. FireVision Detection Model

The authors propose a new deep learning algorithm for computer vision-based fire and smoke detection systems called the FireVision model. The core problem identified is that smoke often envelops

 doi:10.20944/preprints202601.1534.v1

or contains fire in real-world scenarios, leading to significant visual overlap between these classes during detection. This spatial co-occurrence leads to increased classification errors, as the model struggles to distinguish between overlapping fire and smoke regions, thereby elevating the overall loss function that guides the model's training. The increased loss reflects both misclassifications and reduced confidence in predictions at the intersection of fire and smoke, presenting a fundamental obstacle for single-model approaches that attempt to detect both cases simultaneously. To address this challenge, the researchers adopted a specialized architectural approach inspired by prior work on the cooperation of multiple sequential deep learning models [52]. Given that fire is often contained within smoke, the overlap between the two objects significantly increases the classification loss, resulting in a substantial rise in the overall loss. Consequently, the authors developed two distinct models: one exclusively for smoke detection and another specifically for fire detection, effectively decomposing the Firevision process into two stages. Figure 2, illustrates the Firevision detection approach.

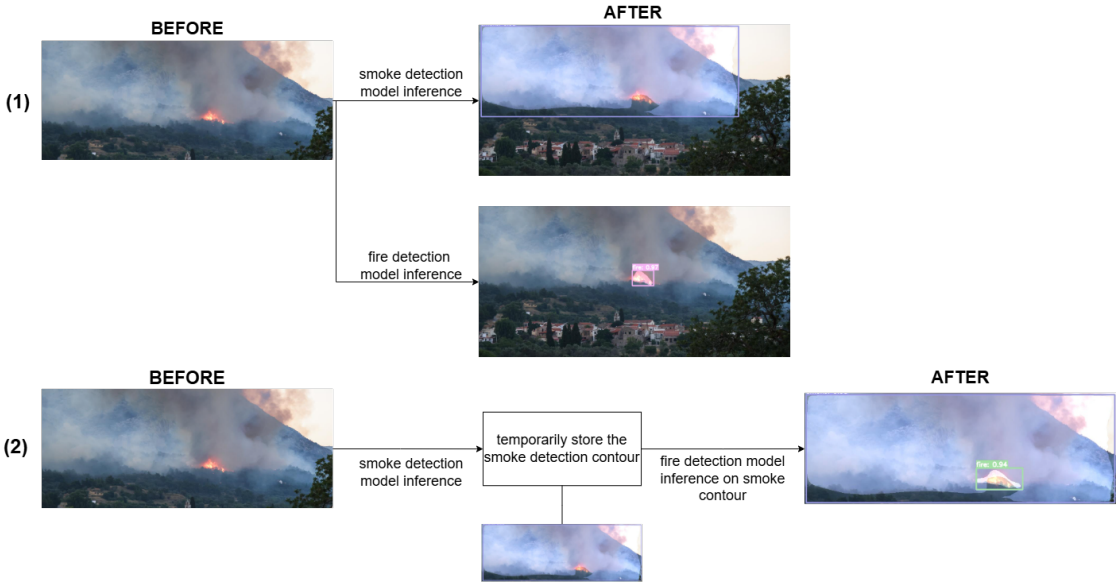

Figure 2. Two Stage Fire and Smoke Detection Algorithm

The Firevision implementation utilizes separate detection models and a two-stage strategy, one dedicated to smoke detection and another to fire detection. This architectural decision effectively transforms a multi-class detection problem into: 1. either two parallel detection tasks of fire and smoke, or 2. sequential detection tasks, initiating the smoke detection and, on the detected smoke contours above a confidence level, initiating the fire detection model. This way, the Firevision model improves precision by eliminating inter-class confusion while maintaining the ability to detect both fire and smoke in a scene through a model ensemble process.

The Firevision implementation models for fire and smoke in this design use Mask R-CNN for detection, with different ResNets for classification (ResNet-50, 101, 152). The classifier model selection for the FPN depents on the drone capabilities in terms of GPU installed coprocessor RAM for the edge computing case and for the cloud case it depends on the inference time and network delay for the object detection results to reach the drone and perform approipriate actions which is considered real-time critical with execution times no more than 2 sec. for the inference Since the study involves comparing three different ResNet architectures (likely ResNet-50, ResNet-101, and ResNet-152 variants), and each architecture must be instantiated twice (once for fire detection, once for smoke detection), the evaluation framework requires assessing six distinct models total. This doubled evaluation burden highlights both the increased resource requirements of the Firevision approach and the comprehensive nature of the experimental design, which follows, which aims to determine not only whether separate models improve detection but also which ResNet variant performs best for each specific detection task.

 doi:10.20944/preprints202601.1534.v1

3.3. Performance Metrics and Loss Functions Used

This section presents the evaluation metrics and loss functions employed to analyze the training behavior of the proposed model and to assess its detection and segmentation performance. The metrics used in this study were selected based on commonly adopted definitions in the literature [81–83]. The examined precision(accuracy) metrics are presented in Table 2, while the loss measures are presented in Table 3.

Table 2. Precision metrics used in the proposed FireVision model experimentation

Metric / Loss	Description
mIoU ∣ IoU	IoU quantifies the overlap between the predicted and ground-truth masks to evaluate localization at pixel level or bounding box overlaps accordingly
Mask ∣ Bbox mAP	Average Precision computed at a fixed mIoU threshold of 0.5, reflecting detection performance under a lenient instance segmentation criterion than COCO mask AP 0.5 . . . 0.95. It matches predicted masks to ground-truth masks using mask IoU. It differs from mAP@0.5 Bbox, used by Faster-RCNN algorithms, which utilizes the bounding box IoU criterion

Table 3. Loss functions monitored in the proposed FireVision model experimentation

Metric / Loss	Description
Classification Loss	Cross-entropy loss quantifies the discrepancy between the predicted class probability distribution and the ground-truth distribution
Bounding Box Regression Loss	Smooth L1 loss measuring discrepancies between predicted and ground-truth bounding box locations
Mask Loss	Pixel-wise binary cross-entropy loss evaluating segmentation accuracy of predicted masks for positive region proposals
Objectness Loss	Binary cross-entropy loss used by the Region Proposal Network to distinguish object regions from background
RPN Bounding Box Loss	Smooth L1 loss applied to refine anchor-based bounding box predictions generated by the RPN
Total Loss	The sum of all classification, localization, and mask-related loss components. (Equation 5)

The evaluation metrics and loss functions adopted in this study are presented in the experimental scenarios section. In the experimental scenarios, the total loss metric is used and derived according to Equation (5), where the mAP0.5 accuracy metric is used for models' mask accuracy results according to Equation (1).

$$mAP_{@0.5}\mathrm{C}^{\mathrm{mask}} = \frac{1}{|D\mathrm{val}|} \sum_{I \in \mathcal{D}\mathrm{val}} I\left[\mathrm{mIoU}(M\mathrm{pred}, M_{\mathrm{gt}}) \geq 0.5\right] \tag{1}$$

 doi:10.20944/preprints202601.1534.v1

The $mAP_{@0.5}$ denotes the Average Precision computed at IoU threshold set equal to 0.5 and above, where C denotes the class either fire of smoke (background is the default class).$|D_{\text{val}}|$ indicates the total number of images used for evaluation (part of the validation set), and $I[\cdot]$, is the indicator function, which yields 1 if the condition inside the brackets is satisfied and 0 otherwise. The condition $\text{mIoU}(M_{\text{pred}}, M_{\text{gt}}) \geq 0.5$ evaluates whether the mask Intersection over Union between the predicted mask M_{pred} and the ground truth mask M_{gt}, meets or exceeds the 0.5 threshold. Here, the mIoU refers to mask IoU rather than bounding box IoU, computed as the area of overlap between predicted and ground truth segmentation masks divided by their area of union.

Although instance segmentation networks such as Mask R-CNN operate on color images, the evaluation of segmentation accuracy is performed on image binary masks. Therefore, for each detected object, the network predicts a probability map $\hat{M}(x,y) \in [0,1]$, which is converted into a binary mask $M_p(x,y)$, using a threshold of 0.5:

$$M_p(x,y) = \begin{cases} 1, & \hat{M}(x,y) \geq 0.5, \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The ground-truth annotation is also a binary mask $M_g(x,y) \in 0,1$. Then the Mask Intersection over Union (Mask IoU) is defined according to Equation (3):

$$mIoU = \frac{|M_p \cap M_g|}{|M_p \cup M_g|} = \frac{\sum_{x,y} M_p(x,y)\, M_g(x,y)}{\sum_{x,y} \left[M_p(x,y) + M_g(x,y) - M_p(x,y) M_g(x,y)\right]} \tag{3}$$

The sequential detections of smoke and fire mAP values are aggregated to an accuracy mAP0.5 value according to Equation (4). The model employs separate detection pipelines for smoke and fire, each producing independent mAP scores. These sequential detection results are aggregated into a unified accuracy metric through a weighted harmonic mean, accounting for the relative detection priorities:

$$mAP_{@0.5;agg} = \frac{(w_f + w_s) \cdot mAP_f \cdot mAP_s}{w_f \cdot mAP_s + w_s \cdot mAP_f} = 2 \cdot \frac{mAP_{@0.5}f \cdot mAP@0.5s}{mAP@0.5f + mAP@0.5_s} \tag{4}$$

where $mAP_{@0.5}f$ and $mAP@0.5_s$ denote the mAP@0.5 scores for fire and smoke detection respectively, and w_f, w_s represent task-specific weighting coefficients. For balanced evaluation, these weights are typically set to $w_f = w_s = 1$, simplifying to the standard F1-score formulation. This aggregation ensures that the combined metric penalizes underperformance in either detection task.

The total loss metric used in the experimental scenarios is a cumulative sum of all losses as shown in Equation (5)

$$\mathcal{L}total = \mathcal{L}cls + \mathcal{L}box + \mathcal{L}mask + \mathcal{L}obj + \mathcal{L}rpn \tag{5}$$

where $\mathcal{L}cls$ is the classification loss for object category prediction, $\mathcal{L}box$ is the bounding box regression loss, $\mathcal{L}obj$ is the objectness loss of the Region Proposal Network, $\mathcal{L}rpn$ is the RPN bounding box regression loss, and $\mathcal{L}_{mask}$ is the pixel-wise mask instance segmentation loss [83].

Frames Per Second (FPS) measures the system's computational performance. In this study, FPS is calculated as the inverse of the total inference time needed to run smoke Mask R-CNN detection, followed by fire Mask R-CNN detection, on a single still image from a drone-mounted camera. For cloud inferences, the measurement input images have a spatial resolution of 5472 by 3648 pixels (20 MegaPixels (MP)), which corresponds to an uncompressed RGB tensor with 8-bit channel depth, approximately 57.1 MB of RAW and 6 to 8 MB of JPEG-compressed images for data transmission. For edge computing inferences, the images have been downscaled to 1280x768 px, to preserve real-time inference performance, since image tiling is not used. These 1MP images, raw size of such images is typically around 3MB (250–900KB JPEG compressed).

 doi:10.20944/preprints202601.1534.v1

If the total time to process one image through both Mask R-CNN models is T_{total} seconds, then FPS is defined according to Equation (6).

$$\text{FPS} = \frac{1}{T_{\text{total}}} = \frac{1}{T_{\text{latency}} + T_{\text{smoke}} + T_{\text{fire}}} \tag{6}$$

where T_{smoke} and T_{fire} are the inference times for the smoke and fire Mask R-CNN models, respectively. The term T_{latency} applies only to the cloud inference cases (not for edge detections), and corresponds to the additional time needed to upload the compressed images of fire and smoke to the cloud. This additional time can be approximated as $T_{\text{latency}} \approx 3$–$6$ seconds per image for 8MB JPEG transmissions over 4G/LTE networks. The approximation used for cloud uploads in this case is the worst-case one and therefore $T_{\text{latency}} \approx 12$ seconds for both smoke and fire images.

3.4. Fire Criticality-Stress Algorithm

Upon object detection using the Firevision model, the proposed Fire criticality-stress algorithm provides a measure of both the severity of the fire event from the acquired images and its expansion, expressed in the event area coverage, derived from the drone's GPS lat-lon values, direct or indirect, as stored in each image's extended information (EXIF). The criticality of a fire-smoke event is expressed by the Fire Criticality Index value (FCI). The criticality index is an aggregative metric that accounts for both fire and smoke detections, as independent contours detected and masked (instance-pixel segmented) by the FireVision model, with the base area being the original image acquired by the drone.

Let an input image of width W and height H have a total pixel area of $A_I = W \cdot H$ (drone original image). Let us assume a set of detected smoke contours indexed by i and fire contours indexed by j. For each smoke detection, let $A^s_{b,i}$ denote the bounding box area of smoke detections and $A^s_{m,i}$ the corresponding Mask-RCNN smoke mask area in pixels. Similarly, for each fire detection, let $A^f_{b,j}$ be the bounding box area for fire detections and $A^f_{m,j}$ the fire mask area. All areas are expressed in pixel units. The total detected smoke and fire mask areas are computed using Equation (7) and are defined as:

$$A^s_m = \sum_i A^s_{m,i}, \qquad A^f_m = \sum_j A^f_{m,j} \tag{7}$$

Similarly, the corresponding total bounding box areas enclosing the contour masks for smoke contours $A^s_{b,i}$ and fire contours $A^f_{b,j}$ are computed using Equation (8).

$$A^s_b = \sum_i A^s_{b,i}, \qquad A^f_b = \sum_j A^f_{b,j} \tag{8}$$

Then the normalized image coverages of smoke C_s and fire C_f are defined using Equation (9):

$$C_s = \frac{A^s_m}{A_I}, \qquad C_f = \frac{A^f_m}{A_I} \tag{9}$$

The authors assume that the normalized smoke and fire coverages satisfy the constraints of Equation (10)

$$C_s < 1, \qquad C_f < 1 \tag{10}$$

If the values are greater than or equal to one, it implies either overlapping contours, duplicated detections, or erroneous mask predictions. If either condition is violated, an iterative correction procedure is applied, in which the largest contour (by mask area) associated with each detection is removed. However, a typical filtering rule is to remove all contours whose area is above 30–70% of the total image area in px^2. This is an expression from dataset experimentation: The smaller the accepted area sizes of the detected contours, the more accurate the detections are. The coverage terms C_s and C_f are then recomputed after each removal step until the constraints in Equation (10) are satisfied, ensuring physically consistent and non-overlapping spatial extents for both smoke and fire detections.

 doi:10.20944/preprints202601.1534.v1

Then, in order to quantify how densely the detected regions are filled by the corresponding masks, the density (fill ratio) terms are introduced and calculated using Equation (11):

$$D_s = \frac{A_m^s}{A_b^s + \epsilon} \qquad D_f = \frac{A_m^f}{A_b^f + \epsilon} \tag{11}$$

where $\epsilon \approx 10^{-6}$, a small positive constant preventing division by zero. The overall fire-event criticality is quantified through the *Fire Criticality Index (FCI)*, defined using Equation (12):

$$\text{FCI} = 100\Big(\alpha\, C_f D_f + \beta\, C_s D_s + \gamma\, C_f C_s\Big), \tag{12}$$

where C_f and C_s denote the normalized over the image fire and smoke coverages, D_f and D_s are the corresponding mask density terms, and α, β, and γ are non-negative weighting coefficients. In this formulation, $\alpha > \beta$ is assumed to reflect the dominant contribution of active fire relative to smoke, while the coefficient γ modulates the interaction between fire and smoke presence. Indicative values that can be used are $\alpha = 0.7$, $\beta = 0.25$, and $\gamma = 0.05$, which significantly prioritize fire extent and yield smaller contour areas for fire relative to smoke, while still accounting for smoke propagation and co-occurrence effects. Values that were used in this work are $\alpha = 0.7$, $\beta = 0.3$, and $\gamma = 0$. The co-occurrence effects are considered when FireVision is applied to fire detection contours inside smoke-detected contours. In case of parallel smoke and fire detections on the original drone image, only α and β coefficients are used.

Equation (12) provides a normalized index FCI $\in [0, 100]$ that is invariant to image resolution and directly comparable across scenes. For example, a small but dense fire occupying 5% of the image ($C_f = 0.05$, $D_f \approx 0.9$) with limited smoke ($C_s = 0.02$, $D_s \approx 0.5$) results in a low-to-moderate criticality value, typically FCI < 20. In contrast, a large-scale event with widespread flames ($C_f > 0.2$) and dense smoke coverage ($C_s > 0.3$), both exhibiting high compactness, produces FCI values exceeding 60, indicating a highly critical fire scenario. This scaling allows the proposed index to distinguish between incipient, developing, and severe fire events in a physically interpretable manner.

A criticality threshold $T_{th} < 3$–5% is used to signify the onset of a fire event. During an automated drone image-acquisition mission along a predefined GPS trajectory from point A to point B, the Fire Criticality Index (FCI) is assessed at the initial anchor A. If $FCI(A) \geq T_{th}$, a secondary exploration mode is triggered. This mode initiates a spiral flight pattern centered at point A to estimate the spatial extent of the affected area in physical units (m^2 or km^2). This process constitutes a core component of the Fire Criticality algorithm, facilitating adaptive, event-driven area-expansion assessment. When the threshold is surpassed, the drone temporarily deviates from its nominal path and follows a piecewise square-spiral trajectory among the predefined straight line flight trajectory from point A to point B, separated by a Euclidean distance d_{AB}. The spiral motion is executed through sequential displacements toward the North, East, South, and West, forming closed square-polygonal coverage regions in polar latitude–longitude coordinates relative to A. The initial spiral step, which determines both the square edge and the forward distance step towards the direction of B, is defined by Equation 13.

$$R_0 = R_d^s = \frac{d_{AB}}{k}, \qquad k \in \mathbb{N},\ k \geq 2, \tag{13}$$

where k is a user-defined control coefficient parameter. The value of k may be either: a) a fixed value R_d^s, following a fixed step square-spiral motion policy or b) an adaptive increase policy starting with a high value of k, let $k_0 = 16$, which is decreased by a value of one, after each full drone square-polygon revolution and forward move (multiplicative increase of R_d^s spiral step). That is, a step length increase-only policy for each step i distance $R_i^s, k = k_i,\ i = 1 \ldots n$, using the Equation (13), and R_0 as a starting step point. Then $\forall i \in \mathbb{N},\ 2 \leq k_i < k_0$, monotonically increase each step's length R_i^s.

Each spiral revolution i consists of a pattern of four directional moves (North, West, South, East), forming a coverage square used for new image acquisitions, detections, and FCI re-evaluations, each time the drone reaches a vertex. If the criticality at a square polygon vertex falls below T_{th}, the spiral

 doi:10.20944/preprints202601.1534.v1

exploration is terminated, and the drone resumes its nominal flight toward point B. Otherwise, the spiral step R_d^s is repeated, either fixed in length or multiplicatively increased, to refine the estimate of the affected area until the termination condition is met. The exact quantity R_d^s is used both as the radial extent of the spiral square around the current anchor and as the forward step length along the $A \to B$ direction.

For example, a drone flies at an altitude h m above ground, and the nominal mission segment has assigned flight position anchors $A \to B$ separated by distance $d_{AB} = 10$ km. If the FCI is above threshold in A, then the spiral descent towards B initiates with a spiral step of the same value for both (i) the spiral-square edge length and (ii) the forward advance distance towards B. This step value is calculated using Equation (14):

$$R_i^s = \frac{d_{AB}}{k_i}, \qquad k_0 = 16,\ i = 1 \dots n \tag{14}$$

where k_i is a selected fixed value for the fixed-length update policy. For the adaptive update policy, the control parameter k_i value is reduced after each completed spiral revolution and forward move so that $k_i = k_{i-1} - 1$. This way, it is guaranteed that the spiral motion is strictly upscaling. (i.e., the next step length is always larger than the previous one). Therefore it is enforced an $R_i^s > R_{i-1}^s$, together with the remaining-distance calculation for $i = n$ to B, $d_n^{rem} \geq R_{i-1}^s$, where $d_{AB_n}^{rem}$, the remaining distance from A to B: $d_{AB_n}^{rem} = d_{AB} - \sum_{i=0}^{n-1} \frac{d_{AB}}{k_i}$. Since an increasing-step policy can become infeasible near the end of the AB segment (when $d_{AB_n}^{rem}$ becomes small), we introduce the *termination condition* close to point B, let it occur at $i = n$:

$$\text{If } d_{AB_{i=n}}^{rem} < R_{i+1}^s = \frac{d_{AB}}{k_{i+1}} \implies \text{terminate spiral exploration and resume nominal flight to } B. \tag{15}$$

If the *termination condition* occurs, then a final spiral revolution may occur for one last time, using a predefined control parameter Δ (fixed value). If this parameter Δ added to $d_{AB_{i=n}}^{rem}$ increases by an amount significant enough to overcome the the limitation imposed from Equation (15), then a last square spiral occurs without forward steping, based on Equation (16):

$$R_{i+1}^s = \min\left(d_{i=n}^{rem} + \Delta,\ R_{i+1}^s\right), \qquad \text{with: } d_{i=n}^{rem} = d_{AB} - \sum_{i=0}^{n} R_i,\ R_{i+1}^s = \frac{d_{AB}}{k_{i+1}} \tag{16}$$

If the $d_i^{rem} + \Delta \geq R_i^s$, then we perform one last square surveilance prior to returning to point B using Equation (17):

$$k_{i+1} = k_i - 1, \qquad R_{i+1}^s = \min\left(\frac{d_{i=n}^{rem} + \Delta}{k_{i+1}},\ \frac{AB}{k_{i+1}}\right) \tag{17}$$

Equation (17) ensures that for the square spiral revolution $i = n$, $R_{n+1}^s \geq R_n^s$, and therefore Equation (15) condition is fulfilled. It also guarantees safe progression toward B, since δ is used only once and prevents Δ from driving the drone beyond B by not implementing the forward step. The algorithm upon $n + 1$ square spiral surveillance completion will not exceed the remaining AB forward, and then it will return directly to B.

Each spiral square revolution i, generates a closed coverage polygon-shape (cartesian coordinates square shape), formed by four directional checkpoints (North, East, South, and West) located at a radial distance R_i from the current location anchor. The resulting polygon is a square box whose edges are both equal to R_i. Therefore, the area of the polygon generated at the n-th revolution is given by Equation (18)

$$A_n^{poly} = \frac{(R_n)(R_n)}{2} = R_n^2 \ \ [m^2]. \tag{18}$$

Assume a drone flies at an altitude $h = 100$ m on a nominal segment from anchor A to anchor B with $d_{AB} = 10$ km and detects a fire incident. Let us assume it is configured to respond using a

 doi:10.20944/preprints202601.1534.v1

fixed-length spiral increase and a fixed control parameter $k = 8$. Then the step is a constant value $R^s = \frac{d_{AB}}{k} = 1250$ m, and $\Delta = 0$. Thus, the drone reaches B exactly after 8 forward steps (revolutions). Let d_i^{rem} be the remaining distance before executing step R_i^s and $d_0^{rem} = d_{AB}$. The fixed-step sequence is:

$Rev(i)$	k	$R_i^s\ (m) = \frac{d_{AB}}{k}$	$d_i^{rem}\ (m)$	$A_i\ (km^2)$
0	8	1250	8750	1.562
1	8	1250	7500	1.562
2	8	1250	6250	1.562
3	8	1250	5000	1.562
4	8	1250	3750	1.562
5	8	1250	2500	1.562
6	8	1250	1250	1.562
7	8	1250	0	1.562

therefore, the mission segment terminates at B after $n = 7$ (eight square revolutions in total).

For the same flight, the adaptive increase policy is performed using Equations (14)–(17), with $\Delta = 250$ m. It initiates with $k_0 = 16$ so that $R_0 = \frac{d_{AB}}{k_0} = 625$ m. After completing the n revolution, we compute the remaining distance $d_n^{rem} = d_{AB} - \sum_{i=0}^{n} R_i^s$, If the termination condition $d_{n+1}^{\text{rem}} \leq R_n^s$ holds. Then the drone performs a final step $R_{n+1}^s = d_{n+1}^{rem}$ and returns to B exactly. Otherwise, the next admissible step is selected by:

$$R_{n+1} = \min\left(d_n^{rem} + \Delta,\ R_{n+1}\right), \qquad \Delta = 250 \text{ m}$$

and mapped to an integer via Equation (17). The resulting sequence until reaching B is:

$Rev(i)$	k_n	$R_i^s\ (m) = \frac{d_{AB}}{k}$	$d_i^{rem}\ (m)$	$A_i\ (km^2)$
0	16	625.0	9375.0	0.39
1	15	666.7	8708.3	0.44
2	14	714.3	7994.0	0.51
3	13	769.2	7224.8	0.59
4	12	833.3	6391.5	0.69
5	11	909.1	5482.4	0.82
6	10	1000.0	4482.4	1.0
7	9	1111.1	3371.3	1.23
8	8	1250.0	2121.3	1.56
9	7	1428.6	692.7	1.77
10	6	1666.6	$692.7 + 250 = 942.7$, direct $\rightarrow B$	–

with a $\Delta = 250$m, the $i = 10$ revolution with $k_i - 1 = 6 = \frac{100000}{6} = 1666.6$m. Therefore the drone will not perform a final tenth revolution and will return to B.

The fixed-step policy ($k = 8$) provides uniform spatial sampling along the trajectory from A to B. This results in identical coverage polygons at each revolution, with a constant area of $A_n = 1.562\ km^2$. This consistency builds trust in systematic scanning and offers clarity in scenarios where fire spread is assumed to be spatially homogeneous. However, this fixed policy does not adjust its spatial resolution based on the event's primary occurrence. This could lead to redundant or excessive coverage in low-risk areas.

In contrast, the adaptive incremental policy dynamically increases the spiral radius and forward step in response to sustained critical conditions of the first fire detection event. This results in progressively larger coverage polygons, ranging from 0.39 km^2 to 1.77 km^2 in the given example. This approach enables rapid expansion of the surveyed area when severe fire conditions persist, while still allowing detailed inspection near the initial triggering location and its neighborhood. Therefore,

 doi:10.20944/preprints202601.1534.v1

the adaptive policy strikes a better balance between local detail and large-scale coverage, making it more suitable for real-time fire-front mapping and area expansion estimation. Additionally, in both cases, the explicit termination rule ensures safe, finite convergence to the destination anchor B, thereby enhancing operational robustness during autonomous missions.

3.5. Dataset and Annotation Process

The data used in this study were obtained from the dataset presented in [84]. This dataset was originally annotated in the YOLO format and could not be used directly in the Firevision training process. For this reason, pixel-level image mask annotations were performed. Using the existing annotations as an initial reference, the images were first categorized into object and no-object classes to facilitate detection, particularly when an object appears very faint or occupies a very small spatial extent within the image. Subsequently, for images known to contain an object of interest, manual annotation was carried out using the LabelMe annotation tool [85]. The overall annotation and dataset preparation workflow is illustrated in Figure 3.

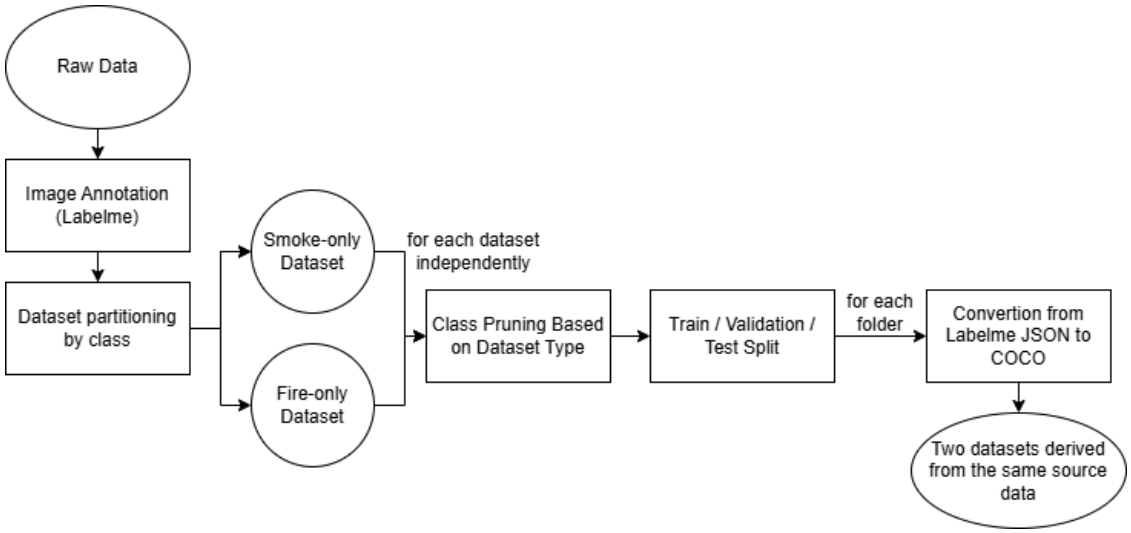

Figure 3. Annotation and dataset preparation workflow.

After completing the annotation process for all images containing objects, the dataset was split into two distinct subsets: one for smoke detection and one for fire detection, each used to train two separate models, as described previously. Following this separation, the non-relevant class was removed from each subset to ensure that the resulting datasets contained exclusively smoke or fire samples, respectively. Each dataset was then divided into training, validation, and test sets, with 70% allocated to training and the remaining 30% split evenly between validation and testing. The corresponding JSON annotation files were included in each split.

Finally, the LabelMe annotation files were converted to the COCO format, which is the annotation standard adopted in this work. Through this process, two task-specific datasets were derived from the original dataset, each tailored to one of the two detection models.

4. Experimental Scenarios

Three experiments have been performed on the FireVision-implemented models and path-planning algorithm during a fire-critical event. The experimental scenario I, aims to evaluate the segmentation accuracy of arbitrary detections from Firevision models and ResNet-50, ResNet-101, and ResNet-152 classifiers for fire and smoke, respectively, using mask mAP@0.5 and total loss metrics as described in section 3.3.

The experimental scenario II involved the use of a drone co-processing GPU delivered by a companion computer unit to provide edge inferences, as simulated by a Jetson Nano device [47,48,79]. The NVIDIA Jetson Nano is a low-power embedded artificial intelligence platform designed for edge-

 doi:10.20944/preprints202601.1534.v1

based deep learning and computer vision applications. This device features a quad-core ARM Cortex-A57 central processing unit (CPU) operating at up to 1.43 GHz and 4 GB of LPDDR4 system memory, shared between the CPU and the graphics processing unit (GPU). The integrated GPU, based on the 128-core NVIDIA Maxwell architecture, delivers up to 472 GFLOPS of computational performance, enabling real-time execution and approximately 25.6 GB/s of memory bandwidth to facilitate high-throughput image processing and neural network inference [46,78]. For cloud-computing inferences, Oracle Cloud Infrastructure (OCI) was used with a minimal VM configuration of 12 x86-64 CPUs and 16GB of memory [86].

Finally, the experimental scenario III provides a proof of concept for the FCI metric, offering inference examples and real FCI values from images in the testing evaluation dataset. It also illustrated the two proposed flight policies that are triggered in the event of a fire incident. The following subsections present the authors' experimental results for the aforementioned scenarios I, II, and III accordingly.

4.1. Experimental Scenario I Results

The training process ran for 100 epochs, with an initial learning rate of 0.005, dynamically adjusted by a CosineAnnealingLR scheduler. Specifically, the learning rate was gradually decreased to 10^{-6} over the first 50 epochs and then held constant for the remainder of the training. The batch size was set to 16 for all training cases, resulting in more stable metric behaviour. The dataset was partitioned and processed following the procedure described in Section 3.5. Table 4 presents the mask map@0.5 and bounding box results for the fire and smoke models, as well as the total loss, classification loss, and mask loss values accordingly.

Table 4. Validation Loss and mAP@0.5 of ResNet backbones for Smoke and Fire Detection

	Smoke Detection model			Fire Detection model		
Measure	ResNet-50	ResNet-101	ResNet-152	ResNet-50	ResNet-101	ResNet-152
Total Loss	0.596	0.575	0.525	1.030	0.925	0.901
Mask Loss	0.380	0.366	0.312	0.435	0.401	0.386
Classification Loss	0.077	0.050	0.042	0.156	0.096	0.089
Mask mAP@0.5	0.574	0.646	0.649	0.409	0.430	0.435
Mask $mAP_{@0.5;agg}$	0.477	0.516	0.52	0.477	0.516	0.52
Bbox mAP@0.5	0.73	0.77	0.785	0.57	0.60	0.61

Evaluation metrics (see Table 4) indicate that ResNet-152 outperforms both ResNet-50 and ResNet-101 in terms of total loss and mAP@0.5. Specifically, for smoke detection models, ResNet-101 reduces the total loss of ResNet-50 by 3.25%, and ResNet-152 achieves an 11.9% reduction compared to ResNet-50. Additionally, the total loss of ResNet-152 is approximately 2% lower than that of ResNet-101. In terms of mask mAP@0.5 values, ResNet-101 demonstrates a 12.54% improvement over ResNet-50. However, ResNet-152 achieves only a 0.46% increase relative to ResNet-101, suggesting that performance gains are approaching saturation.

Similarly, for the fire detection model, the total loss decreases by 10.2% from ResNet-50 to ResNet-101 and by 12.5% from ResNet-50 to ResNet-152. For Mask mAP@0.5 values, a 5.13% increase is observed for ResNet-101 compared to ResNet-50, while the accuracy gains between ResNet-152 and ResNet-101 are only 1.16% increased. Overall, while ResNet-101 exhibits a clear performance advantage over ResNet-50, the incremental gains of ResNet-152 remain limited, especially given its substantially higher computational cost.

 doi:10.20944/preprints202601.1534.v1

To provide a more comprehensive comparison, the authors selecteda recent related study from the literature. In Choi S. et al. [87], Table 6 reports a bbox mAP@0.5 of 0.71 and an instance segmentation mask mAP@0.5 of 0.64 for the Mask R-CNN model with a ResNet-50 backbone on large images. Given that the FireVision models for smoke achieve a bounding-box mAP@0.5 of 0.73 and a mask mAP@0.5 of 0.57, the results are comparable. For the Firevision models of fire, the results are 0.41 for the mask mAP@0.5 and 0.57 for the bounding box mAP@0.5. This is a limitation of the dataset: Fire mask annotations are smaller than smoke annotations. This, of course, leads to significantly lower values of mask mAP@0.5, around 35% lower than the best class results for large objects. The authors note this as a limitation of their annotated dataset, which will be addressed by either enriching it with additional fire images or by super-resolution of existing fire images to provide larger annotated fire contours. Nevertheless, in [87], the comparison using small images gives a mask mAP@0.5 value of 0.107 and a bounding box mAP@0.5 value of 0.143, which is at least 70% smaller than the FireVision fire model's mAP@0.5 values for detection of bounding boxes and masks.

4.2. Experimental Scenario II Results

The experimental scenario II evaluates the cloud and edge instance segmentation inferences of the smoke and fire models. The examined process includes images that present a single contour of smoke detection, followed by a single contour of fire detection. To present the worst-case scenario for edge and cloud computations, the two separate detections apply to the original images. The inference image sizes for cloud inferences are set to high-resolution 20MP images (5472x3648). The inference image sizes for cloud inference are 1280x768, close to 1MP (300–900KB JPEG), to preserve real-time drone inference capabilities. Furthermore, using the same image for 10MP (cloud) and 1MP (edge) detections, the FCI calculations are shown. It should be noted that, due to the memory limitations of the NVIDIA Jetson Nano device, experiments could not be conducted using the same image resolution as in Experimental Scenario 4.1. For this reason, additional benchmark experiments were conducted to examine whether image resolution has a significant impact on the results.

Using the pre-trained models from Scenario 4.1, a similar experimental setup was employed to evaluate the edge performance of the models on the NVIDIA Jetson Nano GPU, in comparison to the performance of the cloud VM. The inference time measurements, the FCI calculation time, and the memory usage for each model are summarized in Table 5.

Table 5. Two stage detection inferences, FCI time, memory usage and FPS of ResNet-50, ResNet-101 and ResNet-152

Backbone Model	Load & Inference Time (sec)	FCI Calculation Time (sec)	Memory Usage (MB)	FPS
Cloud ResNet-50 (5732x3648 px)	4.0	0.02	2555.1	0.062
Cloud ResNet-101 (5732x3648 px)	4.75	0.03	3132.3	0.060
Cloud ResNet-152 (5732x3648 px)	44.6	0.88	6575.6	0.017
Edge ResNet-50 (1280x768 px)	70.42	2.16	1036.9	0.014
Edge ResNet-101 (1280x768 px)	89.64	2.19	1210.2	0.011
Edge ResNet-152 (1280x768 px)	116.48	2.3	1818.5	0.008

 doi:10.20944/preprints202601.1534.v1

According to the results of the total inference time of smoke and fire detection and memory usage reported in Table 5 for both cloud and edge deployment scenarios, several important observations can be drawn. More specifically, in the cloud-based detection scenario, the transition from ResNet-50 to ResNet-101 results in a mean 18.75% increase in inference time, whereas ResNet-152 requires approximately 9x longer inference times than ResNet-101. In contrast, for edge-based detections, the inference time increases by 27.3% from ResNet-50 to ResNet-101, while the difference between ResNet-101 and ResNet-152 show a time increase of approximately 29%. As a result, edge-based fire and smoke detection can no longer be considered real-time, but rather near real-time, making data transmission to the cloud more efficient than performing inference locally on the drone's co-processing unit.

Furthermore, in the cloud-based detection scenario for high-resolution images (20MP), the models' memory consumption reaches particularly high levels, necessitating a reduction in image quality for edge inference, given that the available edge GPU system memory is limited to less than 4GB. Specifically, for the ResNet-152 backbone, edge inference on super-resolution images is not feasible because the memory requirements exceed the available co-processing unit system memory.

In Table 6, the smoke concentration (C_f, see Equation (9)) and the smoke density (D_f, see Equation (11)) measures are evaluated, using the best performing cases in terms of execution time of ResNet-50 and ResNet-101. The results are presented for 1MP (1280x768) and 10MP (5732x3648) images accordingly. The purpose of this evaluation is to estimate detection differences using the same model on images of different resolutions. Therefore, the smoke model is applied to the same image originally captured at 5732x3648 px and then downscaled to 1280x768 px to avoid high memory utilization during inference of high-resolution images (above 3000MB), which can lead to memory exhaustion on the drone edge computing device.

Table 6. Smoke density and concentrations of ResNet-50 and ResNet-101 models using 10MP and 1MP images

Backbone Model	Smoke Density	Smoke Concentration
ResNet-50 (5732x3648 px)	0.67	0.02
ResNet-50 (1280x768 px)	0.66	0.02
ResNet-101 (5732x3648 px)	0.85	0.14
ResNet-101 (1280x768 px)	0.83	0.14

Focusing on the ResNet-50 and ResNet-101 models for cloud and edge model inferences, and concentration and density measures calculated values for each case, it is obvious that high-resolution images do not offer significant changes as expected for these metrics, since the denominator for both measures is the image pixel area for the concentration and the bounding box area for the density measure, which also proportionally decrease on image down-scaling. Furthermore, the lost pixel-area information does not trigger different FCI measure responses when inferred from low-resolution images down to 1280x768 px. Nevertheless, using ResNet-101 instead of ResNet-50 increases the density measure by 30% and, significantly, the concentration measure, regardless of the image resolutions examined.

4.3. Experimental Scenario III Results

Following an FCI event during the drone's predetermined flight from point A to B, the fire criticality-stress index algorithm may utilize two different route path-planning survaelances: one with a fixed step and one with adaptive incremental steps, as illustrated in Figure 4. In this experiment, the original drone course was from its home base H to A (lat=39.805232, lon=20.659988) and then to B (lat=39.783158, lon=20.679147). The straight-line distance AB = 2950.27 m. The Mavlink router service, instantiated on TCP ports 5760 and 5761, was used to communicate with, receive, and send commands to the drone [77], while its course was illustrated using a Google Maps custom UI. The fire notification event was triggered at point A where the FCI value exceeded the threshold value of

 doi:10.20944/preprints202601.1534.v1

0.5. Then, depending on the policy used, the drone automatically performs either a fixed-step-length square spiral move towards B (see Figure 4a) or an adaptive, incremental-step-length square spiral move towards B (see Figure 4b).

(**a**) (**b**)

Figure 4. Example illustration of the automatic drone maneuvers as indicated by the fire criticality algorithm in case of a fire detection event. (a) Fire criticality-stress algorithm deviation of the original drone course in case of a fire incident using a fixed step policy. (b) Fire criticality-stress algorithm deviation of the original drone course in case of a fire incident using an adaptive incremental step policy

For the fixed-step spiral motion, the parameter used was k=5, which led to a fixed step of 590m ($\Delta = 0$). The fixed motion results are presented in Table 7.

Table 7. Square-spiral drone moves. The square edge is of fixed length

Step	Remaining to B (m)	Area (km^2)	Flight coverage (m)
0	2950.2	0.3481	2950
1	2360.1	0.3481	5900
2	1770.1	0.3481	8850
3	1180.1	0.3481	11800
4	590.23	0.3481	**14750.2**

The total coverage area of the fixed-step motion is 1.75 km^2, and the flight coverage in meters the drone actually covered was 14750.2m. For the adaptive incremental step spiral motion, the parameter used was k_0=10, which led to an R_0 step value of 290m. The $\delta = 250m$ was also used in this experiment. The motion results are presented in Table 8.

Table 8. Adaptive square spiral drone moves. The square edge length R is adaptively increased

Step	Remaining to B (m)	R_i^s (m)	Area (km^2)	Drone flight coverage (m)
0	2950.2	295.0	0.0870	1475
1	2655.1	327.8	0.1075	3114
2	2327.3	368.8	0.1360	4958
3	1958.6	421.5	0.1777	7066
4	1537.1	491.7	0.2418	9524
5	1045.4	590.0	0.3481	12474
6	307.9	737.5	05439	15424
-	-	-	-	**15732.9**

The total coverage area of the adaptive increase step motion is 1.64 km^2, and the drone flight coverage in meters was 15732.9m.

The fixed-step square-spiral policy uses a constant square edge of 590 m, resulting in a very regular, predictable scanning pattern. As shown in Table 7, each step covers the same ground footprint,

 doi:10.20944/preprints202601.1534.v1

so the total covered area grows linearly with the number of squares. The cumulative flight distance required to achieve this coverage is 14.75 km, which is relatively efficient because each meter flown contributes almost uniformly to new terrain being scanned. This regularity also simplifies mission planning and battery budgeting, since both coverage growth and energy consumption per step are constant and easy to predict.

In contrast, the adaptive incremental-step policy starts with smaller squares and gradually increases the square edge length as the drone approaches point B. This strategy allows denser inspection near the start of the fire incident, and progressively coarser coverage farther along the path. However, as seen in Table 8, this adaptivity comes at the cost of efficiency: Although the drone flies a longer total distance (15.73 km, about 1 km more than the fixed policy), it covers a smaller total area (1.64 km^2). Therefore, for applications where maximizing surveyed area per unit of flight distance is the primary objective, such as wide-area wildfire monitoring, the fixed-step spiral is the better policy. The adaptive policy is advantageous only when non-uniform spatial resolution is required, for example, when higher inspection density is needed near the starting region or first-time event occurrence, as indicated by the FCI index values.

5. Conclusions

The authors propose a system of automated flights combined with deep learning Mask R-CNN models with ResNet backbones, named FireVision. Two main models are employed by the implementation: One for fire detection and one for smoke detection. Those models can be arbitrarily used either in parallel or sequentially. Furthermore, the system uses indicator measures of fire and smoke phenomena that account for their concentration and density, expressed as image pixels. Combining these two metrics yields a new metric, the Fire Criticality Index (FCI). When the FCI exceeds a predefined threshold, it triggers a modification of the automated flight path using the proposed Fire Criticality Index-stress algorithm. This algorithm aims to estimate the actual surface area of such a critical event, which is subsequently georeferenced to the GPS coordinates of polygon survey areas.

The authors conducted experiments with the FireVision deep learning models in two distinct deployment scenarios: 1) Edge inferences provided by drone co-processing units and 2) cloud inferences. The authors also examined different ResNet backbones, including ResNet-50, 101, and 152, trained on a mask-annotated dataset of 12,000 images, partitioned into 8,500 smoke images and 4,200 fire images. Based on their experiments, the authors identify the ResNet-101 model as the one that offers a high mask map@0.5 accuracy at inference time, making it the candidate model for cloud-based inferences in their FireVision system. For edge-based inference, the authors note the ResNet-50 model's low inference time and the adequate mAP values it achieves compared to the existing literature. Nevertheless, the ResNet-101 model is expected to provide 3—5% more accurate results, at the cost of 27% more inference time, which in some near-realtime cases could be significant enough to endure. Therefore, the ResNet-50 model is the preferred candidate for edge inference, while ResNet-101 is more applicable for more precise cloud inference on high-resolution images, resulting in a significant increase in the FCI index metric when an event is detected.

The authors set as limitations the drones' battery lifetime, which was not examined in the experimental testing of their Fire Criticality algorithm, and the poor mask mAP performance of the fire detection model. The latter will be further exploited by the authors and is set as future work.

Author Contributions: Conceptualization, S.K. and G.K.; methodology, S.K. and G.K.; software, K.S.; validation, K.S, M.T. and S.K. and I.K. and M.K.; formal analysis, K.S. and M.T.; investigation, K.S. and M.T.; resources, S.K. and I.K. and M.K.; data curation, K.S. and M.T.; writing—original draft preparation, K.S. and M.T.; review and editing, S.K. and I.K. and M.K.; visualization, K.S. and M.T.; supervision, S.K. and I.K. and M.K.; project administration, K.S. and M.T. and S.K. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: The authors would like to thank Eleftheria Gramozi, Effrosyni Fotiadou, Ioannis Kostakis, Theofanis Ioannou, Christina Loukakou, Theofilos Loukas and Anna Maria Sidiropoulou, for their valuable contribution to the dataset annotation process.

 doi:10.20944/preprints202601.1534.v1

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Part of the annotated dataset created by the authors and used in this paper to train the deep learning object detection models is available online at [88].

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

Abbreviations

The following abbreviations are used in this manuscript:

AS	application server
CNN	convolutional neural network
CPU	central processing unit
EXIF	exchangeable image file format
FPN	Feature Pyramid Network
GPS	Global Positioning System
R-CNN	Regions with CNN features
RPN	Region Proposal Network
UAV	unoccupied aerial vehicle
UI	User Interface
VM	Virtual Machine
YOLO	You Only Look Once object detection algorithm

References

1. Xu, R.; Yu, P.; Abramson, M.J.; Johnston, F.H.; Samet, J.M.; Bell, M.L.; Haines, A.; Ebi, K.L.; Li, S.; Guo, Y. Wildfires, Global Climate Change, and Human Health. *New England Journal of Medicine* **2020**, *383*, 2173–2181. https://doi.org/10.1056/NEJMsr2028985.
2. Bayham, J.; Yoder, J.K.; Champ, P.A.; Calkin, D.E. The Economics of Wildfire in the United States. *Annual Review of Resource Economics* **2022**, *14*, 379–401. Publisher: Annual Reviews, https://doi.org/10.1146/annurev-resource-111920-014804.
3. Rizzo, L.V.; Rizzo, M.C.F.V. Wildfire smoke and health impacts: a narrative review. *Jornal de Pediatria* **2025**, *101*, S56–S64. https://doi.org/10.1016/j.jped.2024.11.006.
4. Aguilera, R.; Corringham, T.; Gershunov, A.; Benmarhnia, T. Wildfire smoke impacts respiratory health more than fine particles from other sources: observational evidence from Southern California. *Nat Commun* **2021**, *12*, 1493. Publisher: Nature Publishing Group, https://doi.org/10.1038/s41467-021-21708-0.
5. Bolan, S.; Sharma, S.; Mukherjee, S.; Isaza, D.F.G.; Rodgers, E.M.; Zhou, P.; Hou, D.; Scordo, F.; Chandra, S.; Siddique, K.H.M.; et al. Wildfires under changing climate, and their environmental and health impacts. *J Soils Sediments* **2025**, *25*, 3173–3197. https://doi.org/10.1007/s11368-025-04020-y.
6. Pinzon, N.; Galt, R.E.; Roche, L.M.; Schohr, T.; Shobe, B.; Koundinya, V.; Brimm, K.; Powell, J. Farming and ranching through wildfire: Producers' critical role in fire risk management and emergency response. *California Agriculture* **2025**, *79*. https://doi.org/10.3733/001c.128403.
7. Wilner, L.B.; Piepmeier, L.; Gordon, M.; Steiger, B.B.; Northrop, A.J.; McBrien, H.; Shea, B.; Meltzer, G.Y.; Bedi, N.S.; Blake, E.M.; et al. Two and a half decades of United States wildfire burn zone disaster data, 2000-2025. *Sci Data* **2025**, *12*, 1948. Publisher: Nature Publishing Group, https://doi.org/10.1038/s41597-025-06226-8.
8. Modaresi Rad, A.; Abatzoglou, J.T.; Kreitler, J.; Alizadeh, M.R.; AghaKouchak, A.; Hudyma, N.; Nauslar, N.J.; Sadegh, M. Human and infrastructure exposure to large wildfires in the United States. *Nat Sustain* **2023**, *6*, 1343–1351. https://doi.org/10.1038/s41893-023-01163-z.
9. Balch, J.K.; et al. Human-started wildfires expand the fire niche across human-modified landscapes. *Proceedings of the National Academy of Sciences* **2017**, *114*.
10. Tzoumas, G.; Pitonakova, L.; Salinas, L.; Scales, C.; Richardson, T.; Hauert, S. Wildfire detection in large-scale environments using force-based control for swarms of UAVs. *Swarm Intell* **2023**, *17*, 89–115. https://doi.org/10.1007/s11721-022-00218-9.

 doi:10.20944/preprints202601.1534.v1

11. Saleh, A.; Zulkifley, M.A.; Harun, H.H.; Gaudreault, F.; Davison, I.; Spraggon, M. Forest fire surveillance systems: A review of deep learning methods. *Heliyon* **2023**, *10*, e23127. https://doi.org/10.1016/j.heliyon.2023.e23127.
12. Goldammer, J.G.; Mitsopoulos, I.; Byambasuren, O.; Sheldon, P. Defence of Villages, Farms and Other Rural Assets against Wildfires: Guidelines for Rural Populations, Local Communities and Municipality Leaders. Technical report / wildfire protection guidelines, Global Fire Monitoring Center (GFMC), on behalf of the European and Mediterranean Major Hazards Agreement (EUR-OPA), Council of Europe, 2013. [Distributed by the Hellenic Ministry of Climate Crisis and Civil Protection; accessed: 10 April 2023].
13. Edgeley, C.M.; Evans, A.M.; Devenport, S.E.; Kohler, G.; Zamudio, Z.M.; DeGrandpre, W.D. Preventing Human-Caused Wildfire Ignitions on Public Lands: A Review of Best Practices. *For. Sci.* **2025**, *71*, 493–521. https://doi.org/10.1007/s44391-025-00025-9.
14. Barmpoutis, P.; Papaioannou, P.; Dimitropoulos, K.; Grammalidis, N.; Barmpoutis, P.; Papaioannou, P.; Dimitropoulos, K.; Grammalidis, N. A Review on Early Forest Fire Detection Systems Using Optical Remote Sensing. *Sensors* **2020**, *20*. https://doi.org/10.3390/s20226442.
15. Bugarić, M.; Krstinic, D.; Šeric, L.; Stipanicev, D.; Bugaric, M.; Krstinic, D.; Seric, L.; Stipanicev, D. Current Trends in Wildfire Detection, Monitoring and Surveillance. *Fire* **2025**, *8*. https://doi.org/10.3390/fire8090356.
16. Kaur, R.; Singh, S. A comprehensive review of object detection with deep learning. *Digital Signal Processing* **2023**, *132*, 103812. https://doi.org/10.1016/j.dsp.2022.103812.
17. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934* **2020**.
18. Gu, W.; Bai, S.; Kong, L. A review on 2D instance segmentation based on deep neural networks. *Image and Vision Computing* **2022**, *120*, 104401. https://doi.org/10.1016/j.imavis.2022.104401.
19. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2020**, *42*, 386–397. https://doi.org/10.1109/TPAMI.2018.2844175.
20. Zhu, Y.; Wang, X.; Zhang, Z.; Zhang, M. A Comparison of YOLO and Mask-RCNN for Detecting Cells from Microfluidic Images. In Proceedings of the 2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2022, pp. 0998–1002. https://doi.org/10.1109/ICAIIC54884.2022.9722616.
21. Vasconcelos, R.N.; Rocha, W.J.S.F.; Costa, D.P.; Duverger, S.G.; Santana, M.M.M.d.; Cambui, E.C.B.; Ferreira-Ferreira, J.; Oliveira, M.; Barbosa, L.d.S.; Cordeiro, C.L.; et al. Fire Detection with Deep Learning: A Comprehensive Review. *Land* **2024**, *13*. https://doi.org/10.3390/land13101696.
22. Cetinkaya, I.; Catmabacak, E.D.; Ozturk, E. Detection of Fractured Endodontic Instruments in Periapical Radiographs: A Comparative Study of YOLOv8 and Mask R-CNN. *Diagnostics* **2025**, *15*, 653. https://doi.org/10.3390/diagnostics15060653.
23. PyTorch Contributors. Mask R-CNN — Torchvision Models Documentation. https://pytorch.org/vision/stable/models/generated/torchvision.models.detection.maskrcnn_resnet50_fpn.html, 2023. [Online; accessed: 5 September 2024].
24. MultimodalLearning. PyTorch Implementation of Mask R-CNN. https://github.com/multimodallearning/pytorch-mask-rcnn, 2023. [Online; accessed: 5 September 2024].
25. Research, F.A. Detectron2: A PyTorch-based Modular Object Detection Library. https://github.com/facebookresearch/detectron2, 2019. [Online; accessed: 5 September 2024].
26. Yu, H. A Detailed Introduction to ResNet and Its Implementation in PyTorch, 2022. [Online; accessed: 13 March 2023].
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.
28. Kim, Y.; Heo, Y.; Jin, B.; Bae, Y. Real-Time Fire Classification Models Based on Deep Learning for Building an Intelligent Multi-Sensor System. *Fire* **2024**, *7*. https://doi.org/10.3390/fire7090329.
29. Khan, F.; Xu, Z.; Sun, J.; Khan, F.M.; Ahmed, A.; Zhao, Y. Recent Advances in Sensors for Fire Detection. *Sensors* **2022**, *22*, 3310. https://doi.org/10.3390/s22093310.
30. Gomez-Gonzalez, J.L.; Marcoulaki, E.; Cantizano, A.; Konstantinidou, M.; Caro, R.; Castro, M. Simulated fire observables as indicators for optimizing wireless sensor networks in wildfire risk monitoring. *Ecological Indicators* **2025**, *175*, 113509. https://doi.org/10.1016/j.ecolind.2025.113509.
31. Gomez-Gonzalez, J.L.; Marcoulaki, E.; Cantizano, A.; Konstantinidou, M.; Caro, R.; Castro, M. Design of Optimal Wireless Sensor Networks for Enhanced Wildfire Risk Mitigation at Wildland–Human Interfaces.

 doi:10.20944/preprints202601.1534.v1

In Proceedings of the Proceedings of the 35th European Safety and Reliability Conference (ESREL 2025) and the 33rd Society for Risk Analysis Europe Conference (SRA-E 2025), Stavanger, Norway, Jun 2025; pp. 3259—-3265. https://doi.org/10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P6112-cd.

32. Carletti, V.; Greco, A.; Saggese, A.; Vento, B. A Smart Visual Sensor for Smoke Detection Based on Deep Neural Networks. *Sensors* **2024**, *24*. https://doi.org/10.3390/s24144519.
33. NASA EOSDIS. Fire Information for Resource Management System (FIRMS). https://firms.modaps.eosdis.nasa.gov, 2025. [Online; accessed: 10 September 2023].
34. Giglio, L.; Boschetti, L.; Roy, D.P.; Hall, J.V.; Zubkova, M.; Humber, M.; Huang, H.; Oles, V. The NASA VIIRS burned area product, global validation, and intercomparison with the NASA MODIS burned area product. *Remote Sensing of Environment* **2025**, *331*, 115006. https://doi.org/10.1016/j.rse.2025.115006.
35. European Commission.; Joint Research Centre (JRC). European Forest Fire Information System (EFFIS). https://effis.emergency.copernicus.eu, 2025. [Online; accessed:10 April 2025].
36. Xu, W.; Wooster, M.J. Sentinel-3 SLSTR active fire (AF) detection and FRP daytime product - Algorithm description and global intercomparison to MODIS, VIIRS and landsat AF data. *Science of Remote Sensing* **2023**, *7*, 100087. https://doi.org/10.1016/j.srs.2023.100087.
37. Liu, P.; Liu, Y.; Guo, X.; Zhao, W.; Wu, H.; Xu, W. Burned area detection and mapping using time series Sentinel-2 multispectral images. *Remote Sensing of Environment* **2023**, *296*, 113753. https://doi.org/10.1016/j.rse.2023.113753.
38. Pereira, G.H.d.A.; Fusioka, A.M.; Nassu, B.T.; Minetto, R. Active Fire Detection in Landsat-8 Imagery: A Large-Scale Dataset and a Deep-Learning Study. *arXiv* **2021**. arXiv:2101.03409.
39. Schroeder, W.; Oliva, P.; Giglio, L.; Csiszar, I.A. The New VIIRS 375 m active fire detection data product: Algorithm description and initial assessment. *Remote Sensing of Environment* **2014**, *143*, 85–96. https://doi.org/10.1016/j.rse.2013.12.008.
40. Jahagirdar, A.; Sathe, N.; Thorat, S.; Saxena, S. Fire detection in nano-satellite imagery using Mask R-CNN. *International Journal of Signal and Imaging Systems Engineering* **2024**, *13*, 19–26. Publisher: Inderscience Publishers, https://doi.org/10.1504/IJSISE.2024.139980.
41. Ghali, R.; Akhloufi, M.A. Deep Learning Approaches for Wildland Fires Using Satellite Remote Sensing Data: Detection, Mapping, and Prediction. *Fire* **2023**, *6*, 192. https://doi.org/10.3390/fire6050192.
42. Mani, A.; Chen, X.; Gorbachev, S.; Yan, J.; Dixit, A.; Sun, Y.; Yan, Z.; Wu, J.; Deng, J.; Jiang, X.; et al. A Comprehensive Hyperspectral Image Dataset for Forest Fire Detection and Classification. *Scientific Data* **2025**. https://doi.org/10.1038/s41597-025-06404-8.
43. European Space Agency (ESA).; European Commission. Copernicus Hyperspectral Imaging Mission for the Environment (CHIME). https://www.eoportal.org/satellite-missions/chime-copernicus, 2025. [CHIME is designed to provide systematic hyperspectral Earth observations over land and coastal zones to support EU environmental monitoring and natural resource management; accessed December 2025].
44. DJI Enterprise. DJI Manifold 3 User Manual. https://dl.djicdn.com/downloads/Manifold_3/20250828/Manifold_3_User_Manual_v1.2_en.pdf, 2025. [Online; accessed 10 April 2025].
45. DJI Enterprise. DJI Dock 3 and Matrice 4TD Specifications. https://enterprise.dji.com/dock-3/specs, 2025. [Online; accessed 10 April 2025].
46. NVIDIA Corporation. NVIDIA Jetson Nano: Embedded AI Edge Computing Module. https://developer.nvidia.com/embedded/jetson-nano, 2019. [Online; accessed: 30 March 2022].
47. PX4. Companion Computers | PX4 User Guide. https://docs.px4.io/main/en/companion_computer/. [Online; accessed: 20 February 2025].
48. DJI Developer. Jetson Nano Quick Start - Payload SDK (PSDK). https://developer.dji.com/doc/payload-sdk-tutorial/en/quick-start/quick-guide/jetson-nano.html. [Online; accessed: 30 April 2025].
49. Khan, S.; Khan, M.; Hussain, T.; Ser, J.D.; Cuzzolin, F.; Bhattacharyya, S.; Akhtar, Z.; de Albuquerque, V.H.C. DeepSmoke: Deep learning model for smoke detection and segmentation in outdoor environments. *Expert Systems with Applications* **2021**, *182*, 115125. https://doi.org/10.1016/j.eswa.2021.115125.
50. Zhao, Y.; Zhang, H.; Zhang, X.; Chen, X. Fire smoke detection based on target-awareness and depthwise convolutions. *Multimed Tools Appl* **2021**, *80*, 27407–27421. https://doi.org/10.1007/s11042-021-11037-1.
51. Sathishkumar, V.E.; Cho, J.; Subramanian, M.; Naren, O.S. Forest fire and smoke detection using deep learning-based learning without forgetting. *fire ecol* **2023**, *19*, 9. https://doi.org/10.1186/s42408-022-00165-0.
52. Sisias, G.; Konstantinidou, M.; Kontogiannis, S. Deep Learning Process and Application for the Detection of Dangerous Goods Passing through Motorway Tunnels. *Algorithms* **2022**, *15*. https://doi.org/10.3390/a15100370.

53. Shmelkov, K.; Schmid, C.; Alahari, K. Incremental Learning of Object Detectors without Catastrophic Forgetting. In Proceedings of the Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 3400–3409. https://doi.org/10.1109/ICCV.2017.368.
54. Kontogiannis, S.; Kokkonis, G.; Pikridas, C. Proposed Long Short-Term Memory Model Utilizing Multiple Strands for Enhanced Forecasting and Classification of Sensory Measurements. *Mathematics* **2025**, *13*, 1263. https://doi.org/10.3390/math13081263.
55. Peng, C.; Zhao, K.; Maksoud, S.; Li, M.; Lovell, B.C. SID: Incremental Learning for Anchor-Free Object Detection via Selective and Inter-Related Distillation. In Proceedings of the arXiv preprint, 2020. arXiv:2012.15439.
56. Wang, Z.; Wu, L.; Li, T.; Shi, P. A Smoke Detection Model Based on Improved YOLOv5. *Mathematics* **2022**, *10*. https://doi.org/10.3390/math10071190.
57. Yang, L.; Cheng, Y.; Xu, F.; Li, B.; Li, X. Real-Time Smoke Detection in Surveillance Videos Using an Enhanced RT-DETR Framework with Triplet Attention and HS-FPN. *Fire* **2024**, *7*. https://doi.org/10.3390/fire7110387.
58. Govil, K.; Welch, M.L.; Ball, J.T.; Pennypacker, C.R. Preliminary Results from a Wildfire Detection System Using Deep Learning on Remote Camera Images. *Remote Sensing* **2020**, *12*. https://doi.org/10.3390/rs12010166.
59. He, L.; Zhou, Y.; Liu, L.; Zhang, Y.; Ma, J. Research and application of deep learning object detection methods for forest fire smoke recognition. *Scientific Reports* **2025**, *15*, 16328. https://doi.org/10.1038/s41598-025-98086-w.
60. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988. https://doi.org/10.1109/ICCV.2017.324.
61. Kontogiannis, S.; Konstantinidou, M.; Tsioukas, V.; Pikridas, C. A Cloud-Based Deep Learning Framework for Downy Mildew Detection in Viticulture Using Real-Time Image Acquisition from Embedded Devices and Drones. *Information* **2024**, *15*. https://doi.org/10.3390/info15040178.
62. Talaat, F.M.; ZainEldin, H. An improved fire detection approach based on YOLO-v8 for smart cities. *Neural Comput & Applic* **2023**, *35*, 20939–20954. https://doi.org/10.1007/s00521-023-08809-1.
63. Abdusalomov, A.; Baratov, N.; Kutlimuratov, A.; Whangbo, T.K. An Improvement of the Fire Detection and Classification Method Using YOLOv3 for Surveillance Systems. *Sensors* **2021**, *21*. https://doi.org/10.3390/s21196519.
64. Guede-Fernández, F.; Martins, L.; Almeida, R.V.d.; Gamboa, H.; Vieira, P. A Deep Learning Based Object Identification System for Forest Fire Detection. *Fire* **2021**, *4*. https://doi.org/10.3390/fire4040075.
65. Begum, S.R.; S, Y.D.; V, M.M.S. MASK R-CNN for fire detection. *International Research Journal of Computer Science* **2021**, *8*, 145. https://doi.org/10.26562/IRJCS.2021.V0807.003.
66. Casas, E.; Ramos, L.; Bendek, E.; Rivas, F. Assessing the Effectiveness of YOLO Architectures for Smoke and Wildfire Detection. *IEEE Access* **2023**, *11*, 96554–96583. https://doi.org/10.1109/ACCESS.2023.3312217.
67. Alkhammash, E.H. A Comparative Analysis of YOLOv9, YOLOv10, YOLOv11 for Smoke and Fire Detection. *Fire* **2025**, *8*. https://doi.org/10.3390/fire8010026.
68. He, Y.; Sahma, A.; He, X.; Wu, R.; Zhang, R. FireNet: A Lightweight and Efficient Multi-Scenario Fire Object Detector. *Remote Sensing* **2024**, *16*. https://doi.org/10.3390/rs16214112.
69. Xue, Z.; Kong, L.; Wu, H.; Chen, J. Fire and Smoke Detection Based on Improved YOLOV11. *IEEE Access* **2025**, *13*, 73022–73040. https://doi.org/10.1109/ACCESS.2025.3564434.
70. Zhu, W.; Niu, S.; Yue, J.; Zhou, Y. Multiscale wildfire and smoke detection in complex drone forest environments based on YOLOv8. *Scientific Reports* **2025**, *15*. https://doi.org/10.1038/s41598-025-86239-w.
71. Shang, L.; Hu, X.; Huang, Z.; Zhang, Q.; Zhang, Z.; Li, X.; Chang, Y. YOLO-DKM: A Flame and Spark Detection Algorithm Based on Deep Learning. *IEEE Access* **2025**, *PP*, 1–1. https://doi.org/10.1109/ACCESS.2025.3581968.
72. ArduPilot Development Team. Mission Planner, 2025. [Online; accessed: 22 October 2024].
73. QGroundControl Project. QGroundControl: Drone Control Software. https://qgroundcontrol.com/, 2023. [Online; accessed: 09 September 2024].
74. Meier, L.; Honegger, D.; Pollefeys, M. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 6235–6240. https://doi.org/10.1109/ICRA.2015.7140074.
75. Christof, T. DJI Wi-Fi Protocol Reverse Engineering. Bachelor/master thesis, Johannes Kepler University Linz, 2021. Reverse engineering of DJI proprietary communication protocols (e.g., OcuSync/ Wi-Fi).

 doi:10.20944/preprints202601.1534.v1

76. Koubaa, A.; Allouch, A.; Alajlan, M.; Javed, Y.; Belghith, A.; Khalgui, M. Micro Air Vehicle Link (MAVLink) in a Nutshell: A Survey, 2019, [1906.10641].
77. MAVLink Contributors. *MAVLink Developer Guide*. MAVLink Project, 2025. [Online; accessed 10 March 2022].
78. Swaminathan, T.P.; Silver, C.; Akilan, T. Benchmarking Deep Learning Models on NVIDIA Jetson Nano for Real-Time Systems: An Empirical Investigation. *arXiv preprint arXiv:2406.17749* **2024**.
79. Hakani, R.; Rawat, A. Edge Computing-Driven Real-Time Drone Detection Using YOLOv9 and NVIDIA Jetson Nano. *Drones* **2024**, *8*. https://doi.org/10.3390/drones8110680.
80. Voudiotis, G.; Kontogiannis, S.; Pikridas, C. Proposed Smart Monitoring System for the Detection of Bee Swarming. *Inventions* **2021**, *6*, 87. https://doi.org/10.3390/inventions6040087.
81. Salari, A.; Djavadifar, A.; Liu, X.; Najjaran, H. Object recognition datasets and challenges: A review. *Neurocomputing* **2022**, *495*, 129–152. https://doi.org/10.1016/j.neucom.2022.01.022.
82. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the European Conference on Computer Vision (ECCV). Springer, 2014, pp. 740–755. https://doi.org/10.1007/978-3-319-10602-1_48.
83. Terven, J.; Cordova-Esparza, D.M.; Romero-González, J.A.; Ramírez-Pedraza, A.; Chávez-Urbiola, E.A. A comprehensive survey of loss functions and metrics in deep learning. *Artificial Intelligence Review* **2025**, *58*, 195. https://doi.org/10.1007/s10462-025-11198-7.
84. de Venancio, P.V.A.B.; Lisboa, A.C.; Barbosa, A.V. An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices. *Neural Computing & Applications* **2022**, *34*, 15349–15368. https://doi.org/10.1007/s00521-022-07467-z.
85. Wada, K. Labelme: Image Mask Annotation Tool, 2025. [Online; Accessed 15 March 2023], https://doi.org/10.5281/zenodo.5711226.
86. Oracle Cloud Infrastructure Documentation. Compute Shapes in Oracle Cloud Infrastructure, 2024. [Online; accessed 10 March 2024].
87. Choi, S.; Song, Y.; Jung, H. Study on Improving Detection Performance of Wildfire and Non-Fire Events Early Using Swin Transformer. *IEEE Access* **2025**, *13*, 46824–46837. https://doi.org/10.1109/ACCESS.2025.3528983.
88. Microcomputer Systems Laboratory team. Pixel Annotated Dataset for smoke and fire. https://sensors.math.uoi.gr:3002/MCSL_Team/FireAI, 2025. [Online; accessed: 12 November 2025].