

Article

Fire Detection and Geo-Localization Using UAV's Aerial Images and Yolo-Based Models

Kheireddine Choutri ^{1,*}, Mohand Lagha ¹, Souham Meshoul ^{2,*} , Mohamed Batouche ², Farah Bouzidi ¹ and Wided Charef ¹

¹ Aeronautical Sciences Laboratory, Aeronautical and Spatial Studies Institute, Blida 1 University, Blida 0900, Algeria; laghamohand@univ-blida.dz (M.L.)

² Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia; mabatouche@pnu.edu.sa

* Correspondence: choutri.kheireddine@univ-blida.dz (K.C.); sbmeshoul@pnu.edu.sa (S.M.)

Abstract: The past decade has witnessed a growing demand for drone-based fire detection systems, driven by escalating concerns about wildfires exacerbated by climate change, as corroborated by environmental studies. However, deploying existing drone-based fire detection systems in real-world operational conditions poses practical challenges, notably the intricate and unstructured environments and the dynamic nature of UAV-mounted cameras, often leading to false alarms and inaccurate detections. In this paper, we describe a two-stage framework for fire detection and geo-localization. The key features of the proposed work included the compilation of a large dataset from several sources to capture various visual contexts related to fire scenes. The bounding boxes of the regions of interest were labeled using three target levels, namely fire, non-fire, and smoke. The second feature was the investigation of YOLO models to undertake the detection and localization tasks. YOLO-NAS was retained as the best performing model using the compiled dataset with an average mAP50 of 0.71 and an F1_score of 0.68. Additionally, a fire localization scheme based on stereo vision was introduced, and the hardware implementation was executed on a drone equipped with a Pixhawk microcontroller. The test results were very promising and showed the ability of the proposed approach to contribute to a comprehensive and effective fire detection system.

Keywords: UAV; deep learning; stereo vision; YOLO models; Pixhawk; geo-localization; fire detection



Citation: Choutri, K.; Lagha, M.; Meshoul, S.; Batouche, M.; Bouzidi, F.; Charef, W. Fire Detection and Geo-Localization Using UAV's Aerial Images and Yolo-Based Models. *Appl. Sci.* **2023**, *13*, 11548. <https://doi.org/10.3390/app132011548>

Academic Editors: Andrea Prati and Stefano Frizzo Stefenon

Received: 13 September 2023

Revised: 13 October 2023

Accepted: 19 October 2023

Published: 21 October 2023



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

1. Introduction

Fire detection is a critical component of fire safety and prevention, playing a pivotal role in mitigating the devastating consequences of fires. Fires, whether in residential, industrial, or natural settings, pose a significant threat to life, property, and the environment. Rapid and accurate detection of fires is essential to enable timely response measures that can minimize damage, save lives, and safeguard valuable resources. Forest fires have emerged as a significant issue, exemplified in the context of various countries such as Algeria, the USA, and Canada in 2023. The prevalence of combustible materials like shrublands and forests, coupled with a climate conducive to ignition and rapid spread, contributes to the widespread occurrence of fires [1]. Numerous fires ravaged the Mediterranean vegetation, resulting in the destruction of valuable olive trees. The scarcity of water resources posed challenges for distant villages in combating the fires. Some residents fled their homes, while others made valiant attempts to control the flames using rudimentary tools, such as buckets and branches, as the availability of firefighting aircraft was limited. Regrettably, this catastrophe stands as one of the most severe in recent memory [2,3].

The need for effective fire detection systems has become increasingly apparent in recent years due to various factors, including the rising frequency and severity of wildfires, climate change-induced environmental shifts, and the expanding urban landscape. As fires continue to emerge as a global concern, the development and deployment of advanced fire

detection technologies have gained considerable attention. Historically, various methodologies have been employed to detect forest fires, including smoke and thermal detectors [4,5], manned aircraft [6], and satellite imagery [7]. However, each of these techniques carries its limitations, as highlighted in [8]. For instance, smoke and thermal sensors necessitate proximity to the fire, failing to determine the fire's dimensions or location. Ground-based equipment might have restricted surveillance coverage, while the deployment of human patrols becomes impractical in expansive and remote forest areas. Satellite imagery, although valuable, falls short in detecting nascent fires due to insufficient clarity and an inability to provide continuous forest monitoring, primarily due to constrained path planning flexibility. The use of manned aircraft entails high costs and demands skilled pilots, who are exposed to hazardous environments and the risk of operator fatigue, as indicated by [9].

In recent times, an extensive body of research has been conducted in the domain of utilizing unmanned aerial vehicles (UAVs) for monitoring and detecting fires [10]. Within the context of forest fire detection endeavors, UAVs have the potential to fulfill several roles. Initially, UAVs were employed to navigate forest areas, capture video footage, and subsequently analyze the recordings to ascertain the occurrence of fires. As technological advancements have unfolded in the UAV domain, cost-effective commercial UAVs have become accessible for a multitude of research ventures. These UAVs are capable of entering high-risk zones, delivering an elevated vantage point over challenging terrain, and executing nocturnal missions devoid of endangering human lives. The integration of UAVs into these operations presents an array of substantial advantages [11].

- They can cover expansive regions, even in various weather conditions.
- Their operational scope encompasses both day and night periods, with extended mission durations.
- They offer ease of retrieval and relative cost-effectiveness compared to alternative methods.
- In the case of electric UAVs, an additional environmental benefit is realized.
- These UAVs can transport diverse and sizable payloads, catering to varied missions and benefiting from space and weight efficiencies due to the absence of pilot-related safety gear.
- Their capacity extends to efficiently covering extensive and precise target areas.

This paper delves into the realm of fire detection, with a particular focus on the integration of drone technology and vision systems for enhanced capabilities. It explores the challenges posed by fires in complex and dynamic environments, the necessity for reliable detection, and the potential for innovative solutions to address these challenges. By combining cutting-edge technology with data-driven approaches, we aim to improve the accuracy, speed, and adaptability of fire detection systems, ultimately aiming at contributing to more effective fire prevention and response strategies in the face of an increasing fire landscape. In this work, we propose a framework that consists of two main stages for fire detection and localization. The first stage is dedicated to offline fire detection using YOLO (you look only once) models. The second stage is dedicated to online fire detection and localization using the best YOLO model selected in the first stage. The main contributions of this work are as follows.

- We compiled a large dataset that contained more than 12,000 images by using state-of-the-art datasets such as BowFire [12], FiSmo [13], and Flame [14] along with newly acquired images. The compiled dataset included images with scenes documenting several fire and smoke regions.
- We considered three classes, namely fire, non-fire, and smoke, which is an original feature of our work compared to recent works. The non-fire class was used to identify regions that might be mistaken for fire regions because they reflect fire, such as lakes. The dataset has been labeled based on the three mentioned classes.
- The framework applied recent YOLO models to tackle the multi-class fire detection and localization problem.

- We defined the problem of wildfire observation using UAVs and developed a localization algorithm that uses a stereo vision system and camera calibration.
- We designed and controlled a quadcopter based on Pixhawk technology, enabling its real-time testing and validation.

This paper is structured as follows. In Section 2, we survey the existing literature on fire detection using UAVs, fire localization, and vision-based detection methods. In Section 3, we provide an outline of the proposed two-stage framework for fire detection and localization. In Section 4, we describe the methods used in the first stage devoted to offline fire detection using YOLO models. In Section 5, we present the experimental study and results of the first phase. In Section 6, we describe the camera calibration and pose estimation processes for fire detection and localization in a real scenario. The results of the experiments are reported in Section 7. Finally, in Section 8, we draw conclusions and outline plans for future work.

2. Related Work

Our work spans over three main areas, namely using UAVs for fire detection and monitoring, object localization using stereovision, and vision-based models for fire detection and localization. In the following sections, we review the related work in each of these aspects.

2.1. UAVs for Forest Fire Detection and Monitoring

Fire mapping is among the most common tasks in forest fire remote sensing, a process that generates maps indicating fire locations within a specific timeframe using geo-referenced aerial images. Fire maps can also be processed to determine ongoing fire perimeters and estimate positions in unobserved areas. When this mapping process is carried out continuously to provide regular fire map updates, it's referred to as monitoring [15]. Drones play a crucial role in generating comprehensive and accurate fire data through high-resolution cameras, facilitating the characterization of fire geometry. The remote 3D reconstruction of forest fires contributes substantial information for firefighters to assess the fire's severity at specific locations safely [16].

As mentioned in [17], the progression of drone applications within the realm of fire-fighting is primarily focused on the remote sensing of forest fires. Aerial monitoring of such fires is not only expensive but also fraught with significant risks, especially in the context of uncontrolled blazes. The authors in [18] presented an autonomous forest fire monitoring system with the goal of rapidly tracking designated hot spots. Through a realistic simulation of forest fire progression, the authors introduced an algorithm to guide drones toward these hot spots. Their algorithm's performance was then compared against a rudimentary strategy for circling around the fire's existing outline. To test the algorithm, a mixed reality experiment was conducted involving an actual drone and a simulated fire scenario. Drones are poised to perform tasks beyond remote sensing, as discussed in [19], including tasks like aerial prescribed fire lighting. However, the operational implementation of the latter remains underdeveloped due to the logistical challenge of transporting substantial amounts of water and fire retardant. In [20], the authors presented a comprehensive framework utilizing mixed learning techniques, involving the YOLOv4 deep network and LiDAR technology. This cost-effective approach enabled control over the UAV-based fire detection system, which can fly over burned areas and provide precise information. The development of a real-time forest fire monitoring system as described in [21], utilized UAV and remote sensing techniques. Equipped with sensors, a mini processor and a camera, the drone processed data from various on-board sensors and images.

2.2. Stereo Vision-Based Fire Localization System

In [22], a fire location system, utilizing stereo vision, was designed to automatically determine the precise position of a fire and pinpoint the source for extinguishing it. The camera-captured image data underwent analysis via an image processing algorithm to instantly detect the flames. Subsequently, a calibrated stereo vision system was employed

to establish the 3D coordinates of the fire's location or, more accurately, its relative position concerning the camera's perspective. Achieving an accurate fire location stands as a critical prerequisite for facilitating swift firefighting responses and precise water injection.

Predominantly, a stereo vision-based system centered on generating a disparity map was introduced. By computing the camera's meticulous calibration data, the fire's 3D real-world coordinates could be derived. Consequently, an increasing amount of research has delved into fire positioning and its three-dimensional modeling using stereo vision sensors. For instance, the work described in [23] used a stereo vision sensor system for 3D fire location, successfully applying it within a coal chemical company setting. The authors in [24] harnessed a combination of a stereo infrared camera and laser radar to capture fire imagery in smoky environments. This fusion sensor achieved accurate fire positioning within clean and intricate settings, although its effective operational distance remained limited to under 10 m. Similar investigations were conducted in [25], constrained by the infrared camera's operational range and the stereo vision system's base distance. The system proved suitable only for identifying and locating fires at short distances. The authors in [26] established a stereo vision system with a 100 mm base distance for 3D fire modeling and geometric analysis. Outdoor experiments exhibited reliable fire localization when the depth distance was within 20 m. However, the stereo vision system for fire positioning encountered challenges, as discussed in [27,28]. The calibration accuracy significantly influenced the light positioning outcomes, with the positioning precision declining as distance increased. Moreover, the system's adaptability to diverse light positioning distances remained limited. Thus, multiple techniques and solutions have been proposed and adopted to enhance the outcomes.

2.3. Vision-Based Automatic Forest Fire Detection Techniques

The merits of vision-based techniques, characterized by their real-time data capture, extensive detection range, and easy verification and recording capabilities, have positioned them as a focal point in the forest fire monitoring and detection arena [29]. Over the last ten years, research endeavors have extensively employed vision-based UAV systems for fire monitoring and detection. The authors in [30] introduced an innovative method optimized for smart city settings, harnessing YOLOv8 to enhance fire detection precision. In [31], a comprehensive overview explored object detection techniques, focusing on YOLO's evolution, datasets, and practical applications. In [32], the authors refined the detection of forest fires and smoke, offering potential benefits for wildfire management. The work described in [33] advanced forest fire detection reliability using the Detectron2 model and deep learning strategies. Moreover, in [34], the authors suggested enhancing forest fire inspection using UAVs, emphasizing accurate fire identification and location.

Given the complex and unstructured nature of forest fires, leveraging multiple information sources from diverse locations is imperative. When dealing with large-scale or multiple forest fires, a single drone's update rate might be insufficient. The common features used in existing works include the color, motion, and geometry of detected fires. Color, particularly extracted using trained networks, is used to segment fire areas. In [35], the authors introduced a novel framework combining the color-motion-shape features with machine learning for fire detection. Fire characteristics, extending beyond color to irregular shapes and consistent movement at specific spots, were considered. The authors in [36] presented a multi-UAV system for effective forest fire detection. Their paper described in detail the components of the helicopter UAV, offering a vision-based forest fire detection technique that utilized color and motion analyses for clear-range images. The study also showcased algorithms for ultraviolet and visual cameras. A robust forest fire detection system was proposed in [37], necessitating the precise classification of fire imagery against non-fire images. They curated a diverse dataset (Deep-Fire) and employed VGG-19 transfer learning to enhance the prediction accuracy, comparing several machine learning approaches. Furthermore, a deep learning-based forest fire detection model was presented in [38], capable of identifying fires using satellite images. Utilizing RCNNs to

decrease the prediction time, the model achieved a 97.29% accuracy in discerning fire from non-fire images, focusing on unmonitored forest observations.

3. Outline of the Proposed Framework

In this paper, we proposed a fire detection and geo-localization scheme, as shown in Figure 1, where a two-stage framework is shown. The ultimate goal of the fire detection and localization system using UAV images was the timely identification and precise spatial localization of fire-related incidents. The whole process started at the first stage with data collection and integration. Many data sources were used to compile a large dataset of images that represented various visual contexts. The second key component of the proposed framework was the data preparation, which included data augmentation, labeling, and splitting. Then, a third critical component of the system entailed the utilization of innovative YOLO models for object detection, specifically geared towards recognizing three primary classes: fire, non-fire, and smoke. These YOLO models were trained using the labeled images from the data preparation phase in order to identify fire-related objects amidst complex visual contexts. The best performing YOLO model was then used in the second stage of the proposed framework to detect and localize various areas in the images captured by UAVs equipped with high-resolution cameras.

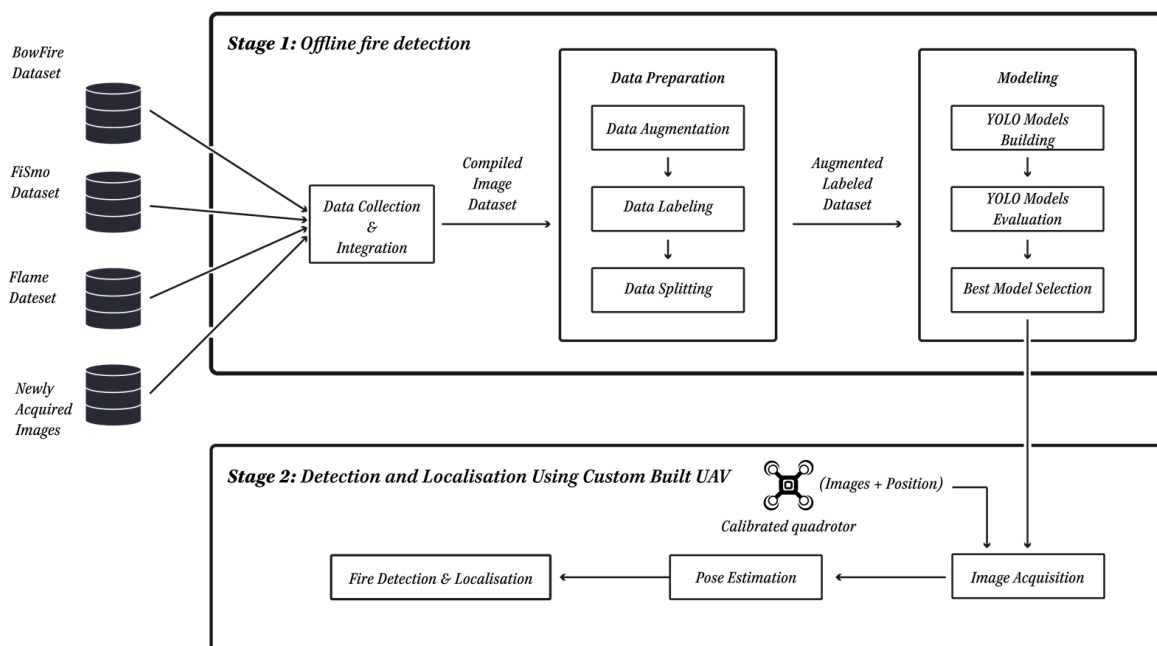


Figure 1. Fire detection and geo-localization proposed framework.

The final and arguably most vital component of the system revolved around precise fire localization, achieved using advanced stereo vision techniques. This stage comprised several interrelated processes. Initially, a meticulous calibration process was conducted to precisely align the UAV's cameras and establish their relative positions. This calibration was foundational, serving as a cornerstone for the subsequent depth estimation. The depth estimation, achieved through stereo vision, leveraged the disparities between the corresponding points in the stereo images to calculate the distance to objects within the scene. This depth information, in turn, fed into the critical position estimation step, allowing the system to calculate the 3D coordinates of the fire's location relative to the UAV's perspective with a high degree of accuracy. Collectively, this integrated system offered a comprehensive and sophisticated approach for fire detection and localization. By harmonizing the data collection from UAVs, YOLO-based object detection, and advanced stereo vision techniques, it not only identified fire, smoke, and non-fire objects but also precisely pinpointed their spatial coordinates. This precision is paramount for orchestrating swift

and effective responses to fire-related incidents, whether in urban or wildland settings, ultimately enhancing the safety of lives, property, and the environment.

4. Materials and Methods for Offline Fire Detection

In this section, we focus on the first stage in the proposed framework. During this stage, three main tasks were conducted, namely data collection and integration, data preparation, and modeling.

4.1. Data Collection, Integration, and Preparation

The effectiveness of deep learning models, such as convolutional neural network (CNN) models for forest fire detection, relies on the quality and size of the datasets utilized. A high-quality dataset allows for deep learning models to capture a wider range of characteristics and achieve enhanced generalization capabilities. To accomplish this, a dataset surpassing 12,000 images was gathered from various publicly available fire datasets, such as BowFire [12], FiSmo [13], Flame [14], and newly acquired images. The collected dataset comprised aerial images of fires taken by UAVs in diverse scenarios and with different equipment configurations. Some representative samples from this dataset are shown in Figure 2. Before utilization, a data preparation process was performed during which the images devoid of fires and those where fires were not discernible were discarded. After the data collection and integration, three main tasks were performed to prepare the data.



Figure 2. Representative samples from the compiled dataset with original labels.

- **Data augmentation:** In order to increase the size of the dataset, amplify the dataset's diversity, and subject the machine learning model to an array of visual modifications, data augmentation was performed. This was conducted by applying geometric transformations such as scaling, rotations, and various affine adjustments to the images in the dataset. This approach enhanced the probability of the model in identifying objects in diverse configurations and contours. We ultimately assembled a dataset containing 12,000 images. Upon assembling the dataset, the images were uniformly resized to dimensions of 416×416 .
- **Data labeling:** The crucial labeling task was carried out manually, facilitated by the MATLAB R2021b Image Labeler app. This task was time and effort consuming as each of the images in the dataset underwent meticulous labeling, classifying the regions in them into one of the three categories: fire, non-fire, or smoke. Completing this labeling procedure entailed creating ground truth data that included details about the image filenames and their respective bounding box coordinates.
- **Data splitting:** The dataset containing all the incorporated features was primed for integration into the machine learning algorithms. Yet, prior to embarking on the algorithmic application, it was recommended to undertake data partitioning. A hold-out sampling technique was used for this purpose where 80% of the images were for training and 20% for testing.

4.2. YOLO Models

There have been numerous deep learning models developed for object detection. Models such as region-based convolutional neural networks (RCNNs) detect objects in two steps. They begin by identifying the areas of interest (bounding boxes) where an item is pre-presented. Second, they classify that specific area of interest. YOLO models are one-stage object detectors in which the model only needs to process the input image once to generate the output. YOLO models are a type of computer vision model capable of detecting objects and segmenting images in real time. The original YOLO model was the first to combine the problems of drawing bounding boxes and identifying class labels in a single end-to-end differentiable network. YOLO models have been shown to be faster, more accurate, and more efficient than alternative object identification models that use two stages or regions of interest. In 2015, Redmon et al. presented the first YOLO model and YOLOv3, YOLOv4, YOLOv5, YOLOv6, YOLOv7, YOLOv8, and YOLO-NAS have been released since then in an effort to keep up with the ever-changing computer vision field and improve the models' performance, accuracy, and efficiency. Each YOLO release added new features, approaches, or architectures to address some of the limitations or issues of the previous releases. For example, YOLOv3, which uses the Darknet-53 backbone, improved small object identification by using multi-scale predictions, whereas YOLOv4 combined numerical features from other models to increase the model's speed and reliability. YOLOv4 was built on a CSPDarknet-53 backbone, which is a modified version of Darknet-53 that uses cross-stage partial connections to reduce the number of parameters and improve feature reuse. It also used a PANet neck, which is a feature pyramid network that combines features from different levels of the backbone and uses attention modules to improve feature representation. Ultralytics' YOLOv5 is a PyTorch-based version of YOLO that offers improved performance and speed trade-offs over the previous versions, as well as support for image segmentation, pose estimation, and classification. In 2022, the YOLOv6 model was released, which uses a ResNet-50 backbone and provides cutting-edge performance on the COCO dataset. YOLOv7 is a 2022 model created by the designers of YOLOv4 that uses a CSPDarknet-53 backbone and incorporates a new attention mechanism and label assignment approach. It can also perform tasks such as instance segmentation and pose estimation. YOLOv8 and YOLO-NAS are the most recent models that are considered powerful additions to the YOLO model family [39].

4.2.1. YOLOv8

Ultralytics introduced YOLOv8 in January 2023, expanding its capabilities to encompass various vision tasks like object detection, segmentation, pose estimation, tracking, and classification. This version retained the foundational structure of YOLOv5 while modifying the CSPLayer, now termed the C2f module. The C2f module, integrating cross-stage partial bottlenecks with two convolutions, merges high-level features with contextual information, enhancing detection precision. Employing an anchor-free model with a disentangled head, YOLOv8 processes object detection and location (termed as objectness in YOLOv8), classification, and regression tasks independently. This approach hones each branch's focus on its designated task, subsequently enhancing the model's overall accuracy. The objectness score in the output layer employs the sigmoid function, indicating the likelihood of an object within the bounding box. For class probabilities, the SoftMax function is utilized, representing the object's likelihood of belonging to specific classes. The bounding box loss is calculated using the CIoU and DFL loss functions, while the classification loss utilizes binary cross-entropy. These losses prove particularly beneficial for detecting smaller objects, boosting the overall object detection performance. Furthermore, YOLOv8 introduced a semantic segmentation counterpart called the YOLOv8-Seg model. This model features a CSPDarknet-53 feature extractor as its backbone, replaced by the C2f module instead of the conventional YOLO neck architecture. This module was succeeded by two segmentation heads, responsible for predicting semantic segmentation masks [39]. More details on the YOLOv8 architecture can be found in [39].

4.2.2. YOLO-NAS

Deci AI introduced YOLO-NAS in May 2023. YOLO-NAS was designed to address the detection of small objects, augment the localization accuracy, and improve the performance–computation ratio, thus rendering it suitable for real-time applications on edge devices. Its open-source framework is also accessible for research purposes. The distinctive elements of YOLO-NAS encompass the following.

- Quantization aware modules named QSP and QCI, integrating re-parameterization for 8-bit quantization to minimize accuracy loss during post-training quantization.
- Automatic architecture design via AutoNAC, Deci's proprietary NAS technology.
- A hybrid quantization approach that selectively quantizes specific segments of a model to strike a balance between latency and accuracy, deviating from the conventional standard quantization affecting all layers.
- A pre-training regimen incorporating automatically labeled data, self-distillation, and extensive datasets.

AutoNAC, which played a pivotal role in creating YOLO-NAS, is an adaptable system capable of tailoring itself to diverse tasks, data specifics, inference environments, and performance objectives. This technology assists users in identifying an optimal structure that offers a precise blend of accuracy and inference speed for their specific use cases. AutoNAC accounts for the data, hardware, and other factors influencing the inference process, such as compilers and quantization. During the NAS process, RepVGG blocks were integrated into the model architecture to ensure compatibility with post-training quantization (PTQ). The outcome was the generation of three architectures with varying depths and placements of the QSP and QCI blocks: YOLO-NASS, YOLO-NASM, and YOLO-NASL (denoting small, medium, and large). The model underwent pre-training on Objects365, encompassing two million images and 365 categories. Subsequently, pseudo-labels were generated using the COCO dataset, followed by training with the original 118 k training images from the COCO dataset. At present, three YOLO-NAS models have been released in FP32, FP16, and INT8 precisions. These models achieved an average precision (AP) of 52.2% on the MS COCO dataset using 16-bit precision [40].

5. Fire Detection Results

Before moving to the next stage of the proposed framework, we present the experimental results of the offline stage.

5.1. Classification Evaluation

In this context of object classification, the most commonly utilized metric is the average accuracy (AP), as it provides an overarching evaluation of model performance. This metric gauged the model's accuracy for a specific label or class. Specifically, we focused on calculating the precision and recall for each of the classes "fire", "non-fire" and "smoke" across all the models. From the precision and recall for each class, the macro-average precision and macro average recall were calculated as the averages over the classes' values for these two metrics, respectively. At the model level, three metrics were considered, namely the F1-score, the arithmetic average class accuracy (arithmetic mean for short), and the harmonic average-class accuracy (harmonic mean for short).

Precision stands as a fundamental measure disclosing the accuracy of our model within a specific class. It was calculated, as shown in Equation (1), as the ratio of TP to the sum of TP and FP for each level of the target, that is, fire, non-fire, and smoke.

$$Precision(l) = \frac{TP(l)}{TP(l) + FP(l)} \quad (1)$$

The precision indicated how confident we could be that a detected region predicted to have the positive target level (fire, non-fire, smoke) actually had the positive target level.

Recall, also known as sensitivity or the true positive rate (TPR), indicates how confident we could be that all the detected regions with the positive target level (fire, non-fire, smoke) were found. It was defined as the ratio of TP to the sum of TP and FN , as shown in Equation (2).

$$Recall(l) = \frac{TP(l)}{TP(l) + FN(l)} \quad (2)$$

The mean average precision at an intersection over union (IOU) threshold of 0.5 (mAP50) was also used to assess the performance of the detection. Using mAP50 meant that model's predictions were considered correct if they had at least a 50% overlap with the ground truth bounding boxes.

For the overall performance of the model, three metrics were considered, namely the F1_score, the arithmetic average class accuracy (arithmetic mean for short), and the harmonic average class accuracy (harmonic mean for short). They were calculated as given by the following equations.

$$F1_{score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

$$arithmetic_average_class_accuracy = \frac{1}{3} \sum_{l=1}^3 Recall(l) \quad (4)$$

$$harmonic_average_class_accuracy = \frac{1}{\frac{1}{3} \sum_{l=1}^3 \frac{1}{recall(l)}} \quad (5)$$

5.2. Model Training

The forest fire detection system underwent training using various YOLO models on a platform equipped with a powerful AMD Ryzen™ 5 5600 H (Santa Clara, CA, USA) processor featuring six cores and 12 threads, complemented by an Nvidia RTX 3060 GPU boasting 6 GB of RAM. The optimization process utilized the stochastic gradient method (SGD) to iteratively update the model parameters. As a result, the key hyper-parameters, including the initial learning rate, batch size, and number of epochs, were configured at 0.003, 16, and 200, respectively. Additionally, we applied specific training

strategies designed to expedite convergence and enhance the model accuracy, which were substantiated by the empirical results. Remarkably, the model exhibited convergence and stability at 100 epochs, as depicted in Figure 3, underscoring the effectiveness of the training parameters for the YOLOv8 model. The detection outcomes across various test dataset scenarios are visually presented in Figure 4. As can be seen in the majority of cases, the system effectively identified and classified fire, non-fire, and smoke regions. However, it's noteworthy that certain minute fire objects remained undetected, primarily attributable to their limited discernibility owing to both distance and image resolution constraints.

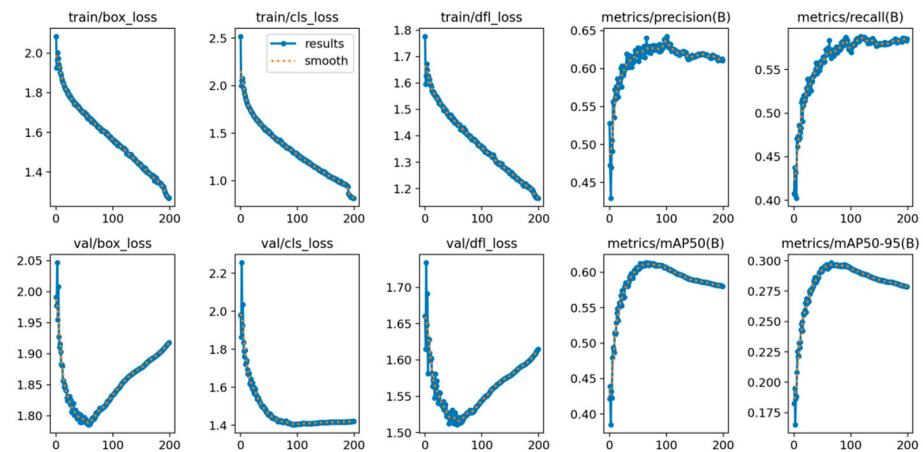


Figure 3. Training metrics using the YOLOv8 detector.



Figure 4. Example of the confidence score diversity using the YOLOv8 detector.

5.3. Comprehensive Comparison of the YOLO Models

Table 1 presents a comparative analysis of the performance metrics for the various models in fire detection, namely YOLOv4, YOLOv5, YOLOv8, and YOLO-NAS. The metrics included the precision (P), recall (R), and mean average precision at IoU 0.5 (mAP50) for the three distinct classes: fire, non-fire, and smoke. The results indicated variations in the models' abilities to accurately detect these fire-related classes.

Table 1. YOLO models evaluation results.

Model	Class	Precision	Recall	mAP50	Macro-avg Precision	Macro-avg Recall	F1-Score	Harmonic Mean	Arithmetic Mean
YOLOv4-tiny	Fire	0.58	0.61	0.66	0.61	0.60	0.61	0.58	0.6
	Non-Fire	0.72	0.71	0.72					
	Smoke	0.54	0.48	0.51					
YOLOv5-s	Fire	0.62	0.66	0.68	0.63	0.62	0.62	0.60	0.62
	Non-Fire	0.73	0.72	0.75					
	Smoke	0.55	0.47	0.50					
YOLOv8-s	Fire	0.64	0.67	0.70	0.67	0.64	0.65	0.62	0.64
	Non-Fire	0.76	0.76	0.83					
	Smoke	0.60	0.50	0.54					
YOLO-NAS-s	Fire	0.67	0.71	0.73	0.70	0.66	0.68	0.63	0.66
	Non-Fire	0.81	0.78	0.87					
	Smoke	0.63	0.48	0.53					

At the class level, YOLOv4 demonstrated reasonable precision for fire (0.58), followed by YOLOv5 (0.62), YOLOv8 (0.64), and YOLO-NAS (0.67). In terms of recall, YOLO-NAS exhibited the highest performance (0.71), closely followed by YOLOv8 (0.67), YOLOv5 (0.66), and YOLOv4 (0.61) for the fire class. Regarding the non-fire class, YOLOv8 outperformed the other models with the highest precision (0.76) and recall (0.76). However, YOLO-NAS achieved the highest mAP50 (0.87), indicating a superior overall performance. YOLOv4, YOLOv5, and YOLOv8 also yielded competitive mAP50 scores (0.66, 0.68, and 0.70, respectively). For the smoke class, YOLOv8 achieved the highest precision (0.60), while YOLOv4 yielded the highest recall (0.50). YOLO-NAS maintained a balanced mAP50 (0.53), suggesting a satisfactory performance in the presence of smoke. Furthermore, Table 1 shows that YOLO-NAS achieved the highest macro-average precision, recall, and F1-score among the models. These results highlight the trade-offs between precision and recall, with YOLO-NAS demonstrating a balanced performance across the classes, making it a notable choice for comprehensive fire detection.

6. Geo-Localization

After identifying YOLO-NAS as the best performing model for fire identification, we now describe the material related to the second stage of the proposed framework where YOLO-NAS was used as the object detector. We first explain how the camera calibration and depth estimation were performed.

6.1. Calibration Process

The calibration procedure relied on a MATLAB application known as the Stereo Vision Camera Calibrator, which incorporates the OpenCV library along with other libraries frequently employed for intricate mathematical operations and specialized functions on arrays of varying dimensions. These libraries also facilitate data management and storage derived from various calibration processes. The core calibration steps were carried out

using the OpenCV library, renowned for its potent computer vision capabilities, encompassing a range of functions pertaining to calibration procedures and a suite of tools that expedite development.

In this work, the MATLAB toolboxes [41,42] were harnessed. These toolboxes offer intuitive and user-friendly applications designed to enhance the efficiency of both intrinsic and extrinsic calibration processes. The outcomes derived from the MATLAB camera calibration toolbox are presented in subsequent sections. The stereo setup consisted of two identical cameras with uniform specifications, positioned at a fixed distance from each other. As depicted in Figure 5, the chessboard square dimensions, serving as the input for the camera calibration, needed to be known (in our instance, it was 28 mm). Upon image selection, the chessboard origin and X, Y directions were automatically defined. Subsequently, exporting the camera parameters to the MATLAB workspace was essential for their utilization in object localization.

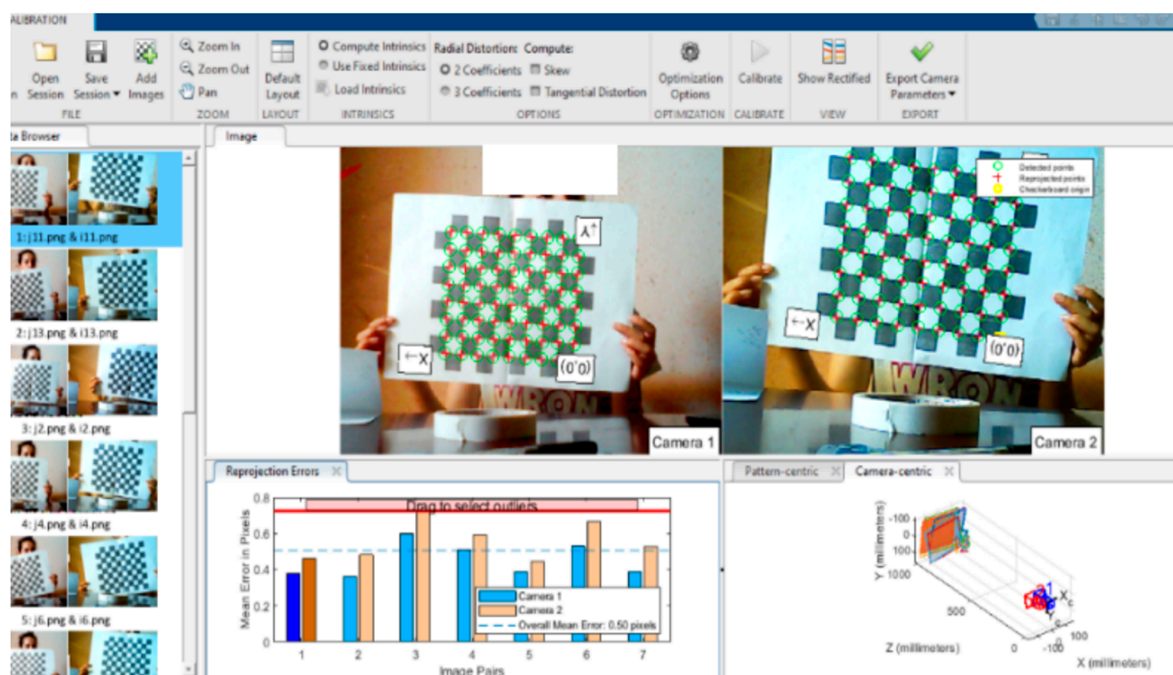


Figure 5. Calibration process.

The outcomes of the simulations yielded the essential camera parameters, such as the focal length, principal point, radial distortion, mean projection error, and the intrinsic parameters matrix. The focal length and principal point values were stored within a 2×1 vector, while the radial distortion was contained within a 3×1 vector. The intrinsic parameters, along with the mean projection error, were incorporated into a 3×3 matrix. These parameters are provided in Table 2.

Table 2. Camera parameters.

Camera	Focal Length	Principal Point	Intrinsic Matrix	Radial Distortion
Camera parameters 1	[932.1819, 929.1559]	[338.5335, 246.8962]	$\begin{bmatrix} 923.1819 & 0 & 0 \\ 0 & 29.1559 & 0 \\ 338.5335 & 246.8962 & 1 \end{bmatrix}$	[0.0509, −1.0153, 20.1072]
Camera parameters 2	$[1.1652 \times 10^3, 1.1760 \times 10^3]$	[355.8581, 330.2815]	$\begin{bmatrix} 1.1652 \times 10^3 & 0 & 0 \\ 0 & 1.1760 \times 10^3 & 0 \\ 335.8581 & 330.2815 & 1 \end{bmatrix}$	[0.0240, −0.1004, 2.4647]

The perspective either centered on the camera or the pattern was designated as camera centric or pattern centric, respectively. This input choice governed the presentation of the camera extrinsic parameters. In our scenario, the calibration pattern was in motion while the camera remained stationary. This perspective offered insights into the inter-camera separation distance, the relative positions of the cameras, and the distance between the camera and the calibration images. This distance information contributed to assessing the accuracy of our calibration procedure.

The re-projection errors served as a qualitative indicator of the accuracy. Such errors represent the disparity between a pattern's key point (corner points) detected in a calibration image and the corresponding world point projected onto the same image. The calibration application presented an informative display of the average re-projection error within each calibration image. When the overall mean re-projection error surpassed an acceptable threshold, a crucial measure for mitigating this was to exclude the images exhibiting the highest error and then proceed with the recalibration. Re-projection errors are influenced by camera resolution and lenses. Notably, a higher resolution combined with wider lenses can lead to increased errors, and conversely, narrower lenses with lower resolution can help minimize them. Typically, a mean re-projection error of less than one pixel is deemed satisfactory. Figure 6 below illustrates both the mean re-projection error per image in pixels and the overall mean error of the selected images.

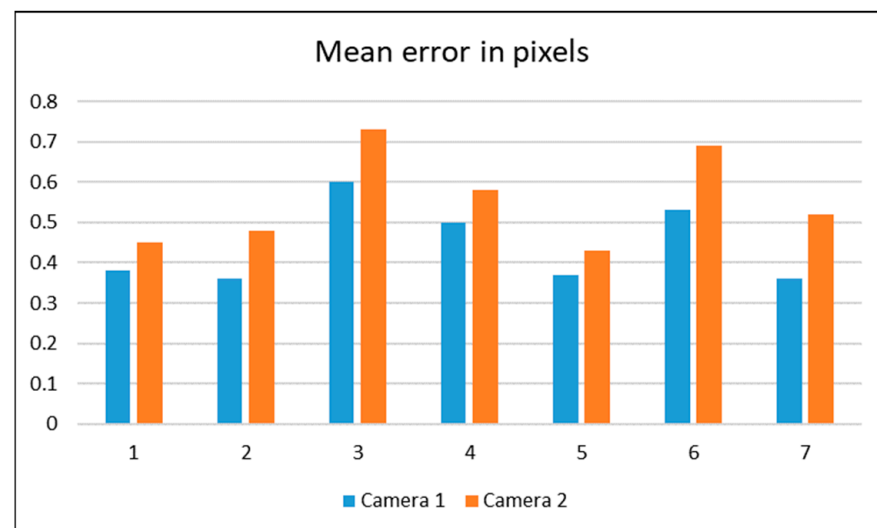


Figure 6. Re-projection error.

6.2. Depth Estimation

Our focus was primarily on fire detection, using it as a case study within our methodology. In our approach, we leveraged the previously discussed trained network to execute the fire detection task, a process that can be succinctly summarized as follows.

The process included the visual representation of the detected object through bounding boxes, as exemplified in Figure 7 below. The process of depth extraction entailed the following sequential steps.

- **Image undistortion:** To enhance the accuracy of the depth estimation, the initial image was subjected to undistortion. This rectified the image deformations resulting from the camera's lenses. The outcome was the undistorted image, which was then employed for fire detection, as depicted in Figure 7.
- **Center coordinates computation:** The center coordinates (X, Y) of each bounding box were calculated.

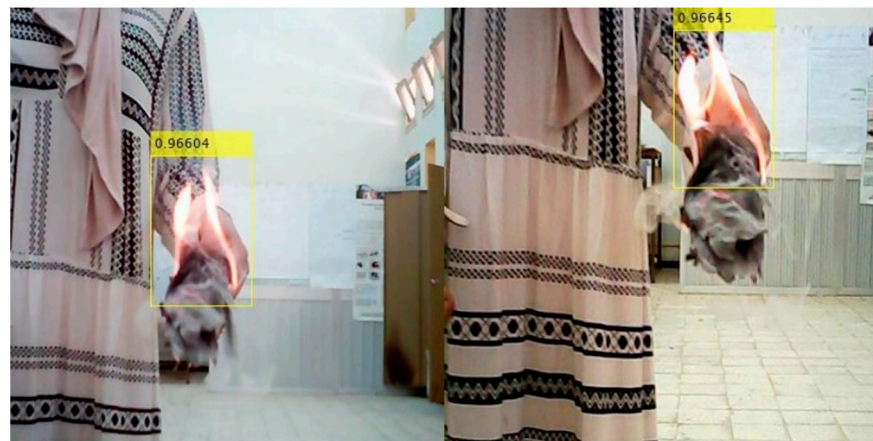


Figure 7. Fire detected in the images adding bounding boxes.

This procedure ensured the extraction of the depth information in a structured manner. As depicted in Figure 8 the depth measurement outcome was recorded as 0.56 m. In comparison, the actual measured distance stood at 0.65 m, resulting in an error of 9 cm. This deviation was within an acceptable range, attributable to factors like the camera resolution, potential calibration discrepancies, and camera orientation.

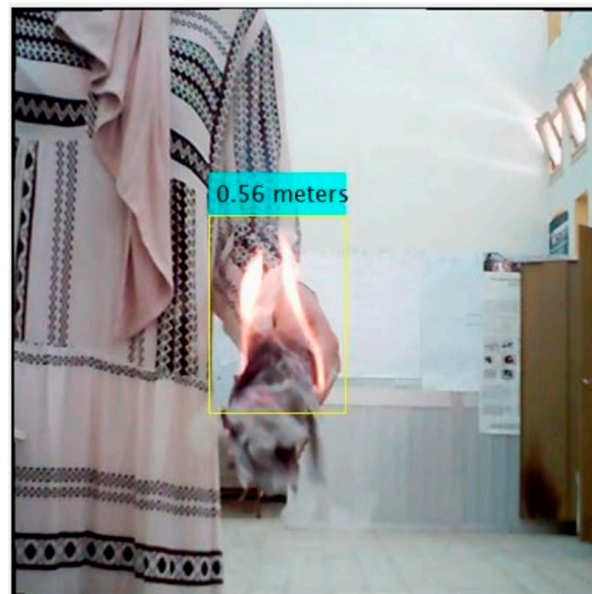


Figure 8. Distance extraction.

6.3. Position Estimation

After successfully extracting the depth information, the subsequent task of calculating the object's relative position to the camera became relatively straightforward, owing to the availability of inverse perspective projection equations as outlined in the earlier sections. The triangulation equations depended on the used camera model. In the case of the pinhole camera model, the equations can be represented as follows.

$$X = \frac{(u - c_x) \cdot Z}{f_x}$$

$$Y = \frac{(v - c_y) \cdot Z}{f_y}$$

where

- (u, v) are the 2D pixel coordinates in the image.
- (c_x, c_y) are the principal point coordinates (intrinsic parameters).
- (f_x, f_y) are the focal lengths along the x and y axes (intrinsic parameters).
- Z is the depth or distance of the point from the camera (the required value).

To obtain the depth or distance (Z) of the point from the camera, the triangulation equations were rearranged and solved for Z using the known values of (u, v) , (c_x, c_y) , (f_x, f_y) , and the calculated values of (X, Y) .

Once the depth Z was determined, the world coordinates (X, Y, Z) of the point were derived. These coordinates represented the position of the point in the world coordinate system.

7. Experimental Results

In this section we report the results we obtained during the second stage of the proposed framework. During this stage, a custom-built UAV was used to acquire images of fire scenes and then YOLO-NAS was used for detecting and locating the region in the images. The depth was then estimated, as explained in the above section.

7.1. Hardware Implementation

For the testing phase, a series of simulations were conducted using our Simulink model. Adjustments were made to the PID controller parameters to fine-tune the performance, yielding the desired outcomes. Subsequently, the model was deployed and loaded onto the PX4 board. Before initiating any flight, it was essential to calibrate the quadcopter meticulously. This process ensured that all the sensors and components were accurately aligned for optimal performance. Furthermore, a thorough safety check was performed to verify the quadcopter's readiness. This entailed inspecting critical aspects, such as the secure mounting of all the propellers and other essential components. The custom-built UAV, as shown in Figure 9, was meticulously configured, and primed for flight, encompassing all the required components described in Table 3.



Figure 9. Final UAV build.

Table 3. Components of the custom-built UAV.

Component	Weight	Characteristics
Frame	282 g	Dimension: 450 mm Voltage distributor: Yes
Motor	52 g	KV (Rpm/V): 2200 Battery operating: 2–3 Lipo Working amps: 14–22 A Thrust at 3S with 1045 propeller: 1200 g

Table 3. *Cont.*

Component	Weight	Characteristics
Electronic Speed Control	35 g	Max current: 40 A Lipo: 2–4 S BEC: 3 A/5 V
Propellers	14 g	Length: 10 inch/254 mm
Battery	243 g	Capacity: 4200 mAh Voltage: 11.1 V Max continuous discharge: 25 c
Pixhawk 2.4.8	40 g	Supply voltage: 7 V Processor: 32 Bit Cortex M4 Core Sensors: Gyrometer, accelerometer, barometer, magnetometer.
Radiolink Receiver R8EF	7 g	Channels: 8 CH Signals: SBUS/PPM, PWM Operating voltage: 3.12 V Control distance: 2000 m

7.2. Fire Detection and Localization

Our experimental focus centered on closely positioned fires, as shown in Figure 10. Specifically, we positioned the fire at the same level as the first camera, allowing us to estimate the preliminary (X, Y, Z) coordinates before embarking on the detailed calculation phase. The outcomes of these tests are meticulously presented in Table 4, highlighting the different instances of relative positioning.

**Figure 10.** Detection test 1.**Table 4.** Object relative position tests.

Test	Coordinates	Camera	World (mm)
1	X	−0.1621	−91.3771
	Y	−0.0312	−17.5973
	Z	-	563.7638
2	X	−0.1219	−386.4144
	Y	−0.0150	−1592.6
	Z	-	3174.3
3	X	−0.1977	−868.1972
	Y	−0.0249	−1704.381
	Z	-	4391.02

In examining the results presented in Table 3, certain observations came to light. It is important to acknowledge that while the calculation of the X- and Y-coordinates provided insights into the object's position, the depth information or Z-coordinate did exhibit some errors. These inaccuracies can be attributed to various factors, including the camera calibration precision and the complexity of the depth extraction. Notably, the objects' positioning within the same horizontal plane as the camera led to smaller X-coordinates. Additionally, the camera's orientation concerning the object contributed to the negative sign in the Y-coordinate, a variance of roughly 15 cm.

Overall, these findings underscored the challenges inherent in accurately determining object positions using stereo vision methods, particularly within a UAV-based application. Further research and refinement of the calibration procedures are essential to enhance the accuracy and reliability of such critical tasks. Furthermore, because of the material limitations, the experiments only focused on detecting fires at short distances. The model, on the other hand, was trained to deal with scenes at far distances. More sophisticated equipment would be more effective in dealing with such situations.

8. Conclusions

A two-stage framework for end-to-end fire detection and geo-localization using UAVs was described in this article. The initial phase was dedicated entirely to offline fire detection and utilized four YOLO models, including the two most recent models, YOLO-NAS and YOLO8, to determine which one was most suitable for fire detection. The models underwent training and evaluation using a compiled dataset comprising 12,530 images, in which regions delineating fire, non-fire, and smoke were manually annotated. The labeling required considerable time and effort. YOLO-NAS emerged as the best performing model among the four under consideration, exhibiting a modest superiority over YOLOv8 in each of the following metrics: precision, recall, F1_score, mAP50, and average class accuracy. YOLO-NAS was implemented in the second stage, which incorporated the analysis of real-life scenarios. In this stage, the images captured by a custom-built UAV were supplied to YOLO-NAS for the purposes of object detection and localization. Geo-localization was also considered by employing accurate camera calibration and depth estimation techniques. The test results were extremely encouraging and demonstrated the overall process's viability, although it could be enhanced in numerous ways. In the future, we intend to use optimization algorithms to fine-tune the hyper-parameters of the YOLO models, specifically YOLO8 and YOLO-NAS, in order to further enhance their performance. Furthermore, more advanced UAVs need to be used to evaluate the system in real-world forest fire settings.

Author Contributions: Conceptualization, K.C. and M.L.; data curation, F.B. and W.C.; methodology, S.M. and M.B.; software, K.C.; supervision, S.M., M.L. and M.B.; validation, M.B., K.C. and S.M.; writing—original draft, K.C.; writing—review and editing, M.B., M.L. and K.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R196), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets are available from the corresponding authors upon request.

Acknowledgments: The authors would like to acknowledge the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R196), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

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

References

1. Choutri, K.; Fadloun, S.; Lagha, M.; Bouzidi, F.; Charef, W. Forest Fire Detection Using IoT Enabled UAV And Computer Vision. In Proceedings of the 2022 International Conference on Artificial Intelligence of Things (ICAIoT), Istanbul, Turkey, 29–30 December 2022; pp. 1–6.
2. Choutri, K.; Mohand, L.; Dala, L. Design of search and rescue system using autonomous Multi-UAVs. *Intell. Decis. Technol.* **2020**, *14*, 553–564. [\[CrossRef\]](#)
3. Choutri, K.; Lagha, M.; Dala, L. A fully autonomous search and rescue system using quadrotor UAV. *Int. J. Comput. Digit. Syst.* **2021**, *10*, 403–414. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Khan, F.; Xu, Z.; Sun, J.; Khan, F.M.; Ahmed, A.; Zhao, Y. Recent advances in sensors for fire detection. *Sensors* **2022**, *22*, 3310. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Yu, J.; He, Y.; Zhang, F.; Sun, G.; Hou, Y.; Liu, H.; Li, J.; Yang, R.; Wang, H. An Infrared Image Stitching Method for Wind Turbine Blade Using UAV Flight Data and U-Net. *IEEE Sens. J.* **2023**, *23*, 8727–8736. [\[CrossRef\]](#)
6. Hao, K.; Wang, J. Design of FPGA-based TDLAS aircraft fire detection system. In Proceedings of the Third International Conference on Sensors and Information Technology (ICSI 2023), Xiamen, China, 15 May 2023; Volume 12699, pp. 54–61.
7. Hong, Z.; Tang, Z.; Pan, H.; Zhang, Y.; Zheng, Z.; Zhou, R.; Ma, Z.; Zhang, Y.; Han, Y.; Wang, J.; et al. Active fire detection using a novel convolutional neural network based on Himawari-8 satellite images. *Front. Environ. Sci.* **2022**, *10*, 794028. [\[CrossRef\]](#)
8. Jijitha, R.; Shabin, P. A review on forest fire detection. *Res. Appl. Embed. Syst.* **2019**, *2*, 1–8.
9. Zhang, L.; Wang, M.; Fu, Y.; Ding, Y. A Forest Fire Recognition Method Using UAV Images Based on Transfer Learning. *Forests* **2022**, *13*, 975. [\[CrossRef\]](#)
10. Partheepan, S.; Sanati, F.; Hassan, J. Autonomous Unmanned Aerial Vehicles in Bushfire Management: Challenges and Opportunities. *Drones* **2023**, *7*, 47. [\[CrossRef\]](#)
11. Ahmed, H.; Bakr, M.; Talib, M.A.; Abbas, S.; Nasir, Q. Unmanned aerial vehicles (UAVs) and artificial intelligence (AI) in fire related disaster recovery: Analytical survey study. In Proceedings of the 2022 International Conference on Business Analytics for Technology and Security (ICBATS), Dubai, United Arab Emirates, 16–17 February 2022; pp. 1–6.
12. Chino, D.Y.; Avalhais, L.P.; Rodrigues, J.F.; Traina, A.J. Bowfire: Detection of fire in still images by integrating pixel color and texture analysis. In Proceedings of the 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images, Salvador, Brazil, 26–29 August 2015; pp. 95–102.
13. Cazzolato, M.T.; Avalhais, L.; Chino, D.; Ramos, J.S.; de Souza, J.A.; Rodrigues, J.F., Jr.; Traina, A. Fismo: A compilation of datasets from emergency situations for fire and smoke analysis. In Proceedings of the Brazilian Symposium on Databases-SBBD, SBC Uberlândia, Minas Gerais, Brazil, 2–5 October 2017; pp. 213–223.
14. Shamsoshoara, A.; Afghah, F.; Razi, A.; Zheng, L.; Fulé, P.Z.; Blasch, E. Aerial imagery pile burn detection using deep learning: The FLAME dataset. *Comput. Netw.* **2021**, *193*, 108001. [\[CrossRef\]](#)
15. Novo, A.; Fariñas-Álvarez, N.; Martínez-Sánchez, J.; González-Jorge, H.; Fernández-Alonso, J.M.; Lorenzo, H. Mapping Forest Fire Risk—A Case Study in Galicia (Spain). *Remote Sens.* **2020**, *12*, 3705. [\[CrossRef\]](#)
16. Bouguettaya, A.; Zarzour, H.; Taberkit, A.M.; Kechida, A. A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms. *Signal Process.* **2022**, *190*, 108309. [\[CrossRef\]](#)
17. Twidwell, D.; Allen, C.R.; Detweiler, C.; Higgins, J.; Laney, C.; Elbaum, S. Smokey comes of age: Unmanned aerial systems for fire management. *Front. Ecol. Environ.* **2016**, *14*, 333–339. [\[CrossRef\]](#)
18. Skeeel, R.C.; Hollinger, G.A. Aerial vehicle path planning for monitoring wildfire frontiers. In *Field and Service Robotics*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 455–467.
19. Beachly, E.; Detweiler, C.; Elbaum, S.; Duncan, B.; Hildebrandt, C.; Twidwell, D.; Allen, C. Fire-aware planning of aerial trajectories and ignitions. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 685–692.
20. Kasyap, V.L.; Sumathi, D.; Alluri, K.; Ch, P.R.; Thilakarathne, N.; Shafi, R.M. Early detection of forest fire using mixed learning techniques and UAV. *Comput. Intell. Neurosci.* **2022**, *2022*, 3170244. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Wesly, U.J.; Chaitanyab, R.V.S.; Kumarc, P.L.; Kumard, N.S.; Devie, B. A Detailed Investigation on Forest Monitoring System for Wildfire Using IoT. In Proceedings of the First International Conference on Recent Developments in Electronics and Communication Systems (RDECS-2022), Surampalem, India, 22–23 July 2022; Volume 32, p. 315.
22. Kustu, T.; Taskin, A. Deep learning and stereo vision-based detection of post-earthquake fire geolocation for smart cities within the scope of disaster management: Istanbul case. *Int. J. Disaster Risk Reduct.* **2023**, *96*, 103906. [\[CrossRef\]](#)
23. Song, T.; Tang, B.; Zhao, M.; Deng, L. An accurate 3-D fire location method based on sub-pixel edge detection and non-parametric 545 stereo matching. *Measurement* **2014**, *50*, 160–171. [\[CrossRef\]](#)
24. Tsai, P.F.; Liao, C.H.; Yuan, S.M. Using deep learning with thermal imaging for human detection in heavy smoke scenarios. *Sensors* **2022**, *22*, 5351. [\[CrossRef\]](#)
25. Zhu, J.; Li, W.; Lin, D.; Zhao, G. Study on water jet trajectory model of fire monitor based on simulation and experiment. *Fire Technol.* **2019**, *55*, 773–787. [\[CrossRef\]](#)
26. Toulouse, T.; Rossi, L.; Akhloufi, M.; Pieri, A.; Maldague, X. A multimodal 3D framework for fire characteristics estimation. *Meas. Sci. Technol.* **2018**, *29*, 025404. [\[CrossRef\]](#)

27. McNeil, J.G.; Lattimer, B.Y. Robotic fire suppression through autonomous feedback control. *Fire Technol.* **2017**, *53*, 1171–1199. [CrossRef]
28. Wu, B.; Zhang, F.; Xue, T. Monocular-vision-based method for online measurement of pose parameters of weld stud. *Measurement* **2015**, *61*, 263–269. [CrossRef]
29. Yuan, C.; Zhang, Y.; Liu, Z. A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned 556 aerial vehicles and remote sensing techniques. *Can. J. For. Res.* **2015**, *45*, 783–792. [CrossRef]
30. Talaat, F.M.; ZainEldin, H. An improved fire detection approach based on YOLO-v8 for smart cities. *Neural Comput. Appl.* **2023**, *35*, 20939–20954. [CrossRef]
31. Diwan, T.; Anirudh, G.; Tembhurne, J.V. Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimed. Tools Appl.* **2023**, *82*, 9243–9275. [CrossRef] [PubMed]
32. Li, J.; Xu, R.; Liu, Y. An Improved Forest Fire and Smoke Detection Model Based on YOLOv5. *Forests* **2023**, *14*, 833. [CrossRef]
33. Abdusalomov, A.B.; Islam, B.M.S.; Nasimov, R.; Mukhiddinov, M.; Whangbo, T.K. An improved forest fire detection method based on the detectron2 model and a deep learning approach. *Sensors* **2023**, *23*, 1512. [CrossRef]
34. Lu, K.; Xu, R.; Li, J.; Lv, Y.; Lin, H.; Liu, Y. A Vision-Based Detection and Spatial Localization Scheme for Forest Fire Inspection from UAV. *Forests* **2022**, *13*, 383. [CrossRef]
35. Harjoko, A.; Dharmawan, A.; Adhinata, F.D.; Kosala, G.; Jo, K.H. Real-time Forest fire detection framework based on artificial intelligence using color probability model and motion feature analysis. *Fire* **2022**, *5*, 23.
36. Sudhakar, S.; Vijayakumar, V.; Kumar, C.S.; Priya, V.; Ravi, L.; Subramaniaswamy, V. Unmanned Aerial Vehicle (UAV) based Forest Fire Detection and monitoring for reducing false alarms in forest-fires. *Comput. Commun.* **2020**, *149*, 1–16. [CrossRef]
37. Khan, A.; Hassan, B.; Khan, S.; Ahmed, R.; Abuassba, A. DeepFire: A Novel Dataset and Deep Transfer Learning Benchmark for Forest Fire Detection. *Mob. Inf. Syst.* **2022**, *2022*, 5358359. [CrossRef]
38. Chopde, A.; Magon, A.; Bhatkar, S. Forest Fire Detection and Prediction from Image Processing Using RCNN. In Proceedings of the 7th World Congress on Civil, Structural, and Environmental Engineering, Virtual, 10–12 April 2022.
39. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8 2023; 2023. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 1 June 2023).
40. Aharon, S.; Dupont, L.; Masad, O.; Yurkova, K.; Fridman, L.; Lkdci; Khvedchenya, E.; Rubin, R.; Bagrov, N.; Tymchenko, B.; et al. Super-Gradients. *GitHub Repos.* **2021**. [CrossRef]
41. Mathwork. Computer Vision Toolbox. 2021. Available online: <https://www.mathworks.com/products/computer-vision.html> (accessed on 1 May 2021).
42. Mathwork. Image Processing Toolbox. 2021. Available online: <https://www.mathworks.com/products/image.html> (accessed on 1 May 2021).

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