

Article

Deep Learning-Based Fire Detector Robust to Smoke–Fog Ambiguity in Outdoor Scenes

Sangmin Suh ^{1,2} 

¹ Department of Information and Telecommunication Engineering, Gangneung-Wonju National University, Wonju-si 26403, Republic of Korea; sangminsuh@gwnu.ac.kr or sangmin.suh@scalable-ai.io

² Scalable-AI, Wonju-si 26403, Republic of Korea

Abstract

In previous studies, fire detection models that only differentiate between fire and smoke are presented. However, a high false detection rate occurs because of confusion between smoke and fog. In this study, a novel method is proposed to effectively distinguish between smoke and fog. A custom dataset is introduced for detecting fire, smoke, and fog, which offers a novel labeling technique to reduce the misclassification of smoke and fog. A new architecture is proposed that is aimed at enhancing the detection performance. The latest You Only Look Once version 11 (YOLOv11) is used to establish a performance baseline. Further research then focuses on improving this benchmark. To ensure accurate texture differentiation, the dataset is designed to exclude overlapping ground-truth boxes, enabling the trained model to determine object boundaries independently, which is a design approach not found in previous studies. The model design aims to maximize performance and cost efficiency. The performance is improved by adding an image pyramid layer to the existing model to improve large-object detection. Cost efficiency is improved by designing a new module, C5Go, to mitigate the additional computational load introduced by the added pyramid level. Comparative experiments on the proposed custom fire dataset demonstrate that the proposed model improves the detection performance while keeping the additional computational overhead modest. The experimental results show that the proposed model achieves a 12.86% improvement in smoke detection performance compared with YOLOv11 and attains an overall mean average precision (mAP)_{@50} score of 0.906, reflecting superior performance. The contributions of this work are as follows: (i) we design a three-class dataset comprising fire, smoke, and fog so as not to cause false detections; (ii) we propose a new model structure to improve performance; and (iii) we verify that the proposed method indeed improves performance through the experimental results.

Keywords: deep learning; fire detection; object detection; real-time detection; smoke detection; YOLOv11



Received: 25 January 2026

Revised: 10 February 2026

Accepted: 14 February 2026

Published: 16 February 2026

Copyright: © 2026 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

According to the Korea Fire and Rescue Service, approximately 40,000 fires have occurred in Korea over the past decade, resulting in approximately 2000 casualties [1]. Despite advances in technology and several different preventive measures, it is difficult to clearly identify the factors that cause fires because fires occur in different forms in our daily lives. This limits the ability to reduce the number of fires. Therefore, it is important to detect the early signs of a fire quickly to minimize the damage. Currently, two main types of fire detectors are used: heat and smoke detectors. Both methods detect the

physical and chemical changes that occur when heat or smoke from a fire reaches the detector sensor. However, this approach is limited by the time that it takes for the heat and smoke to reach the detector sensor, and a possibility of malfunction due to external factors exists. To overcome this problem, studies are being conducted to combine new technologies with sensor data, rather than relying solely on data. Examples include a hybrid fire detection system that combines computer vision and metal oxide semiconductor (MOS) Gas Sensor-2 (MQ-2) smoke sensors [2] and a forest fire detection and warning system using an Arduino ATmega2560-based wireless sensor network and Internet of Things technology [3]. However, sensor-based fire detectors are limited by the detection radii and the time required for the sensors to sense the heat or smoke. Studies that combine drone technology and artificial intelligence are actively underway. Examples include a drone-based forest fire early detection and distance estimation system combining an attention gate (AG) U-Net and Oriented FAST and Rotated BRIEF Simultaneous Localization and Mapping (ORB-SLAM) [4]; a real-time fire detection model named the Active Fire Network (AF-Net) that uses high-resolution images [5]; a real-time forest fire management system using You Only Look Once version 5 (YOLOv5)-based drones [6]; an early fire detection forest monitoring system using convolutional neural networks (CNNs) and unmanned aerial vehicle (UAV) images [7]; and a forest fire identification system using YOLOv4 and aerial images [8]. However, these studies are mainly focused on wildfires, which makes it difficult to apply the results to detect small or remote fires. Moreover, in the case of fire detection using drones, limitations regarding the drone battery capacity and flight time make continuous monitoring difficult. In addition, despite being used in recent studies, it is difficult to evaluate the performance of the older model YOLOv4. In addition, publicly available datasets for wildfire computer vision research have been curated and continuously expanded [9].

To address these limitations, fire detection research that combines image processing and deep learning technologies has been gaining increasing attention. The need for technologies capable of accurately and reliably detecting the early stages of fires in real time is becoming increasingly clear. For instance, a fire detection network using irregular flame feature detection and the visualized heat release rate was used for indoor fire detection [10]. However, this approach did not detect smoke, a precursor to fire, and the authors failed to provide inference times, making it difficult to evaluate its real-time performance. In another approach, an indoor fire load detection model was developed based on a mask region-based convolutional neural network (Mask R-CNN), focused on detecting and segmenting combustible materials that contribute to the fire load in indoor environments [11]. However, the model exhibited low mean average precision (mAP) of 0.405, which is insufficient for detecting small objects or handling complex backgrounds. A study leveraging EfficientNet (EFFNet), a multitask network for video smoke detection and source prediction, emphasized smoke detection but lacked sufficient consideration of real-world flames [12]. Related video-based fire recognition methods have also been subjected to local/global adaptive enhancement to improve visibility and robustness [13]. A CNN-based deep learning model was proposed to automate the classification of multisource fires [14]. Although innovative, this study prioritized fire source analysis over detection and did not effectively analyze the interaction between smoke and flames because it focused primarily on classification tasks [14]. Efforts have also been made to explore fire detection using different color spaces. These include approaches such as a Fast R-CNN integrated with color-based anchoring and global information networks [15]; a real-time fire detection system using OpenCV and hue, saturation, and value (HSV) color models [16]; and image- and video-based real-time fire detection models using the HSV and YCbCr color spaces [17]. Other methods have used red, green, and blue (RGB) and hue, saturation,

and intensity (HSI) color models [18] or developed fire classification systems based on RGB colors and support vector machines [19].

Recently, studies have focused on improving fire detection performance using deep learning-based models. Examples include studies combining CNN and long short-term memory models to analyze spatial and temporal features [20]; a multiscale fire detector based on an improved YOLOv4 [21]; the GAM-ASFF-YOLOv5 + REP model that enhances YOLOv5 for detecting flames and smoke [22]; fire detection algorithms based on YOLOv5 [23]; lightweight real-time wildfire detection models using YOLOv5 [24]; a YOLOv7-based wildfire detection system with edge computing [25]; an early fire detection model leveraging the Spatio-Temporal Self-Attention Network (STNet) [26]; and a fire detection model based on DETection TRansformer (DETR) [27]. However, most of these approaches focus primarily on detecting fires, without addressing the detection of smoke, which is a critical fire indicator. Moreover, because a fast response is required in real-world deployment, lightweight and computationally efficient fire detection models are essential. Real-world deployment also requires compact models with low computational overhead, yet many studies do not consistently report efficiency-related metrics in a comparable manner (e.g., model complexity or runtime settings).

To enable early fire detection, it is crucial to identify smoke as an indicator. However, detecting smoke poses challenges owing to its unstructured nature, lack of distinctive features, and strong visual similarity to natural phenomena such as fog. The use of computer vision techniques that rely on image processing presents a significant risk of false positives when fog is misclassified as smoke. Several fire/smoke object detection datasets have been released in recent years [28]; however, most studies do not explicitly model fog as a hard-negative class.

To address these challenges, this study proposes a method to clearly distinguish between fog and smoke, thereby reducing false positives. The proposed approach detects smoke and fire and incorporates a fog class to prevent misclassification, minimizing the unnecessary deployment of firefighting resources. The latest YOLOv11 framework was used for the base model, and several new architectural enhancements were introduced to improve its performance. The effectiveness of the proposed method is demonstrated through a comparative analysis with existing models, demonstrating its superiority. The highlights of this study are given as follows:

- A novel fire dataset is provided to distinguish between smoke and fog;
- A modified image pyramid structure is proposed to enhance the recognition of unstructured objects;
- A ghost block architecture is introduced to achieve a lightweight model for cost efficiency;
- For a fair performance evaluation, the same fire dataset is applied to existing models, and their performance is compared.

The structure of this paper is as follows. Section 2 describes the proposed datasets and model architecture. Section 3 presents the training results and performance evaluation, and Section 4 concludes the study.

2. Robust Fire Detector Design

This section contains key details regarding the proposed dataset and model design, which are central to the study.

2.1. Dataset and Label Design

In existing fire/smoke detection research and actual outdoor environments, cases of fog being falsely detected as smoke are frequently reported. This is because the two phenomena share similar visual textures and low contrast characteristics. In this

study, fog is defined as a representative hard-negative condition from the perspective of smoke detection and is thus a separate class. Therefore, this study aims to alleviate smoke–fog confusion and reduce false positives in smoke detection by separating fog into a separate class, thereby improving the overall fire early detection performance.

To create the dataset, we utilized images from public datasets, and the labels were redesigned for use in our model. Several open datasets for fire object detection have been developed. D-Fire [29] is a dataset including outdoor scenes; the Flame And Smoke Detection Dataset (FASDD) [30] is a benchmark dataset for fire and smoke object detection; BoWFire [31] is a relatively small dataset for fast training and evaluation; and Roboflow [32] is a data platform containing various types of images and annotations. Among them, we used the FASDD and the Roboflow data in this study. The total number of images in the datasets was 3100, and the numbers of instances of fire, smoke, and fog were 1896 (37.8%), 1758 (35.0%), and 1360 (27.1%), respectively.

Object detection, unlike image classification, requires both images and bounding box coordinates. This involves drawing bounding boxes to indicate the locations of objects within an image and assigning appropriate class labels to compile a dataset. This process is called labeling, and a labeling tool can be used to adjust annotations to fit a specific data format.

A consistent labeling criterion is essential. This study deals with unstructured data, such as fire, smoke, and fog, which have irregular shapes and often overlap. To eliminate potential confusion caused by overlapping objects, this study ensures that the bounding boxes do not overlap. Non-overlapping annotations do not eliminate coexistence but rather reduce the label noise resulting from texture mixing in overlapping regions, thereby providing clear textures for each class. However, for example, in a scene with strong overlap between fire and smoke, if only one class is conservatively annotated, the model may interpret visual cues in the overlapping region as the other class, potentially increasing class confusion between fire and smoke. The applied approach helps the model to clearly distinguish textures during training, enabling the trained model to make predictions based on the distinctive characteristics of each texture. Table 1 presents the class names and their corresponding class IDs, while Figure 1 provides examples from the dataset.



Figure 1. Designed datasets. (a) Fire and smoke labels. (b) Fog label.

When creating the dataset, images were resized using the letterbox method to minimize the distortion of the original images. This approach preserved the aspect ratio while

transforming the original images to a specific size. Consequently, the dataset contained images with zero padding, as shown in Figure 1.

Table 1. Class names and indices.

Class ID	Class Name
0	fire
1	smoke
2	fog

LabelMe was used as the labeling tool, generating JSON format files for each image. As the YOLO model requires label files in text format, a preprocessing step was performed to convert the JSON files into text format. The entire dataset consisted of 3100 images, which were split into training (train), validation (valid), and test (test) sets in a 7:2:1 ratio. Specifically, the dataset was divided into 2170 training images, 620 validation images, and 310 test images.

2.2. Proposed Model Design

To design an effective model, we first identified the shortcomings of existing models and then created a new architecture that addressed the identified limitations. We selected the recent YOLOv11 [33] as a baseline model and, in particular, employed the YOLOv11s variant—which is well suited for rapid inference tasks, such as fire detection—to conduct experiments and establish baseline performance.

Experiments using native YOLOv11 repeatedly revealed that, among the fire, smoke, and fog classes, the detection performance for the smoke class was the lowest. Analysis indicated that smoke exhibits distinct characteristics—it tends to be large, has diffuse boundaries, and is distributed throughout the entire image—making it particularly challenging for conventional models to detect. More specifically, difficulty arises because the relative scale and morphology of smoke differ substantially from those of typical object categories in general-purpose object detection benchmarks. Moreover, models that are optimized for small-object detection struggle to accurately detect such large objects. To address this issue, we enhanced the large-object detection capabilities by adding an extra layer at the top of the existing image pyramid. This addition naturally necessitated modifications to the neck structure. In the proposed neck architecture, we expanded the conventional concatenation process to fuse feature maps from multiple scales. While the original YOLOv11 model employs three output scales (20×20 , 40×40 , and 80×80), our approach improves upon this by utilizing four output scales (10×10 , 20×20 , 40×40 , and 80×80). This modification optimizes the model for the more effective detection of both small and large objects, as confirmed by our experiments. The augmented image pyramid structure is illustrated in Figure 2.

However, this additional image pyramid inevitably increases the computational cost and the number of parameters. In this study, we propose a novel module called C5Go to replace C3K2 in order to partially offset the additional computation introduced by the P6 layer; its architecture is shown in Figure 3.

Importantly, C5Go is applied at the fifth level rather than the sixth level of the image pyramid. This design choice was made to decouple the effect of adding an extra pyramid level from the benefits introduced by the new module. The proposed C5Go is designed to reduce the computational cost associated with the conventional C3K2 module, and its detailed structure is presented in Figure 4.

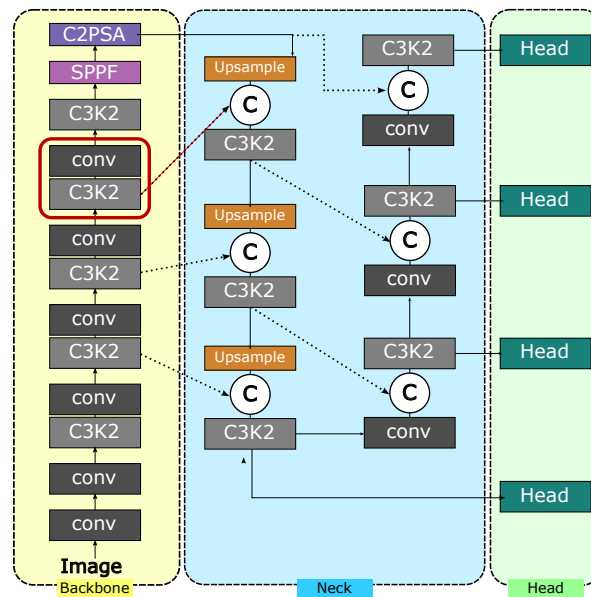


Figure 2. Structure with an additional P6.

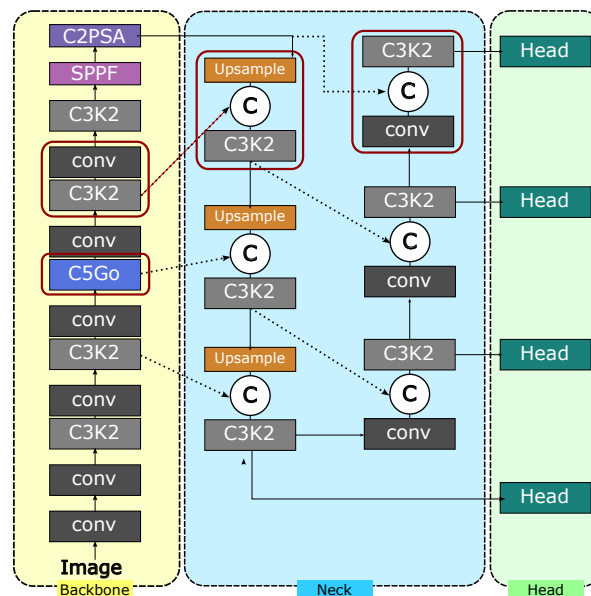


Figure 3. Structure with an additional P6 and C5Go.

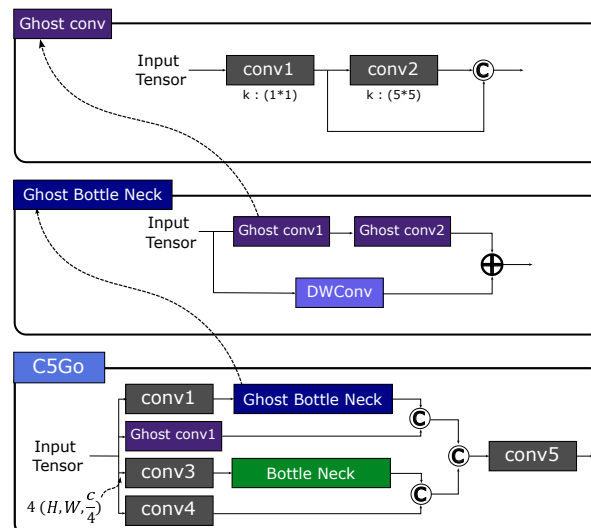


Figure 4. C5Go structure.

C5Go consists of a total of five convolutional operations, four of which utilize conventional convolution, while the remaining one employs ghost convolution to optimize the computational cost. In CNN models, the computational cost of convolution operations (measured in FLOPs, or floating point operations) is a crucial factor that determines the model's size and efficiency. The FLOPs for a standard 2D convolution operation are defined as follows:

$$\text{FLOPs}_{\text{conv}} = C_i \times C_o \times K^2 \times H \times W, \quad (1)$$

where C_i denotes the number of input channels, C_o represents the number of output channels, K is the kernel size of the conventional convolution, and H and W refer to the height and width of the output feature map, respectively. GhostConv reduces redundant feature map generation in CNN operations and leverages depthwise convolution to maintain performance while reducing the computational cost. The FLOPs for GhostConv are computed as follows:

$$\begin{aligned} \text{FLOPs}_{\text{ghost}} &= C_i \times \frac{C_o}{r} \times K^2 \times H \times W \\ &= \left(1 - \frac{1}{r}\right) \times K_g^2 \times H \times W, \end{aligned} \quad (2)$$

where r is the ghost ratio, which denotes the proportion of the total output channels that are computed directly using standard convolution, and K_g represents the kernel size of the depthwise convolution. As can be seen from the above equation, GhostConv reduces the computational cost by a factor of $1/r$ compared to conventional convolution, while the remaining feature maps are generated using depthwise convolution, thereby lowering the overall computational expense. In summary, in this study, we mitigate the increased computational cost associated with the original YOLOv11 by employing GhostConv for some of the convolution operations within the C5Go module, all while maintaining comparable performance. This approach not only enhances the computational efficiency but also enables the optimal utilization of memory and multiply–accumulate (MAC) resources.

3. Performance Evaluation

The training of the final model incorporating the proposed P6 and C5Go, as designed in the previous section, was carried out using the hyperparameters as shown in Table 2.

Table 2. Hyperparameters.

Parameter Name	Value
batch_size	32
epochs	200
learning rate	1×10^{-2}
dropout	0.2
optimizer	stochastic gradient descent (SGD)
learning rate scheduler	cosine annealing

To ensure training stability, we employed a cosine annealing learning rate schedule and set the dropout rate to 0.2 to prevent overfitting. In addition, model optimization was performed using stochastic gradient descent (SGD) to maintain optimal performance. Figure 5 illustrates the evolution of the loss function during training. The graph shows that

the loss gradually decreases as training progresses, indicating that the model is learning in a stable manner.

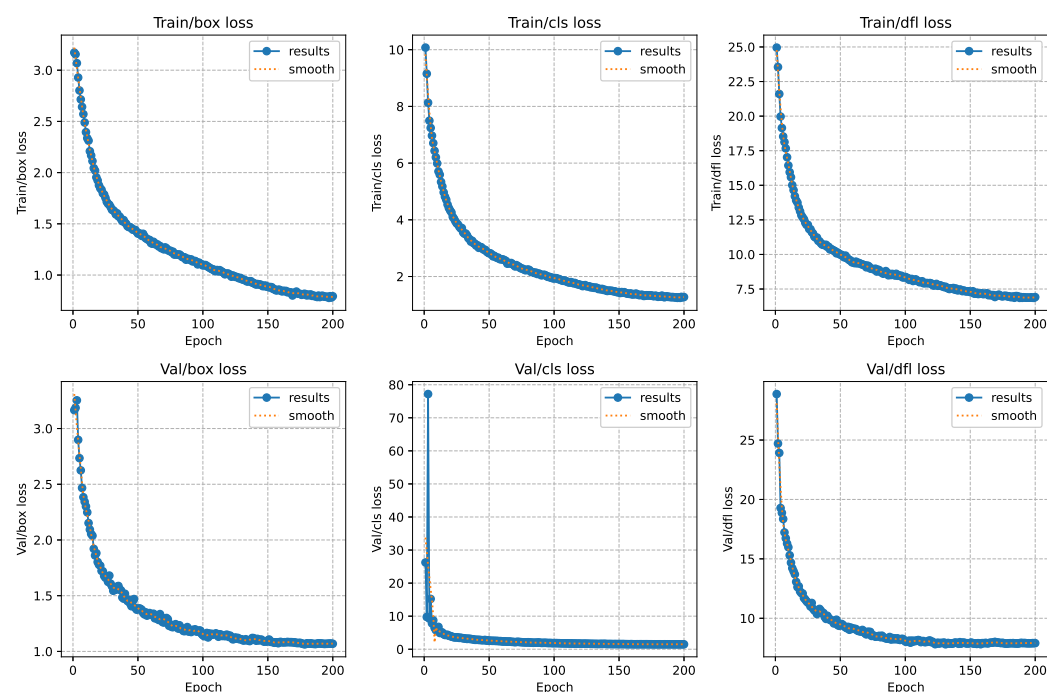


Figure 5. Training and evaluation losses.

To evaluate the performance of the object detection model, the mean average precision (mAP) is used in this study. The mAP@50 (mean average precision at IoU 0.5) is the average precision calculated with a fixed intersection over union (IoU) threshold of 0.5. This metric is useful for rapidly evaluating the overall detection performance and is commonly used as a baseline performance indicator in object detection models. The mAP@50-95 (mean average precision at IoU 0.5 to 0.95) is computed by varying the IoU threshold from 0.5 to 0.95 in increments of 0.05, calculating the average precision (AP) for each of the 10 IoU thresholds, and then averaging these values. This metric provides a more granular and stringent evaluation criterion, making it well suited for assessing both the precision and generalization performance of models. Therefore, in this study, both metrics were employed to comprehensively analyze model performance. In this work, the baseline model exhibited poor performance in smoke detection, prompting the proposal of a new architecture to address this issue. To verify the improvements in detail, we conducted class-specific comparative experiments. Table 3 presents the mAP results obtained when using the original YOLOv11 model, while Table 4 shows the results obtained with the proposed architecture.

Table 3. Conventional YOLOv11 performance.

	Fire	Smoke	Fog	Total
mAP@50	0.88	0.75	0.98	0.876
mAP@50-95	0.58	0.49	0.812	0.632

Table 4. Proposed model performance.

	Fire	Smoke	Fog	Total
mAP@50	0.902	0.822	0.994	0.906
mAP@50-95	0.614	0.553	0.798	0.655

Recall that the purpose of this study was to alleviate smoke–fog ambiguity by explicitly modeling fog as a hard-negative class in the detection framework, thereby reducing false positives in smoke detection and improving the overall early fire detection performance. Based on the mAP@50-95, the improvements for fire, smoke, and fog were +5.8%, +12.86%, and −1.72%, respectively. This indicates that the detection rate for smoke is improved the most, with a modest improvement in fire detection and nearly unchanged performance for fog. Therefore, the substantial improvement in smoke detection achieved by the proposed architecture demonstrates its excellent suitability for early fire detection and suppression.

Figure 6 shows the confusion matrix of the designed detector, which implies that confusion between smoke and fog is negligible. The remaining FP/FN errors are largely attributable to confidence-thresholding effects and can be mitigated by lightweight post-processing, e.g., class-wise thresholds.

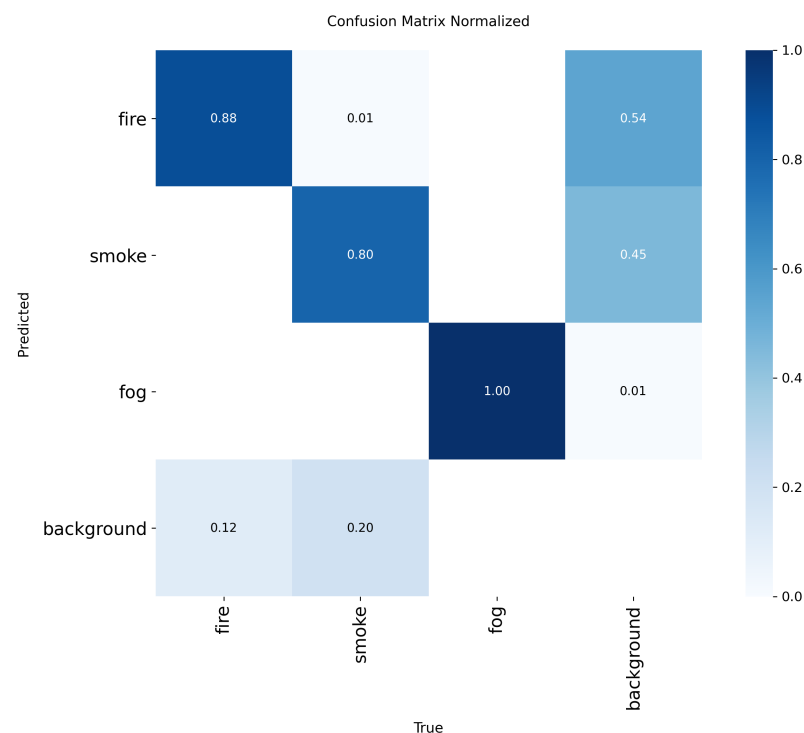


Figure 6. Normalized confusion matrix.

While the quantitative gains depend on the deployment domain, the underlying mechanism of our approach is not dataset-specific: modeling fog explicitly as a hard-negative class directly targets a common real-world failure mode of smoke detectors under foggy/hazy conditions. In practical deployment, the remaining background-related FP/FN errors can be further reduced using lightweight operating point calibration, such as class-wise confidence thresholds and simple temporal filtering for video streams. The main contribution of this study is the proposal of a general design method to prevent detection errors between fog and smoke classes. However, the quantitative performance variation may vary depending on variations in camera, lighting, and weather conditions, and further fine-tuning may be necessary.

In addition, using YOLOv11 as the baseline model, we analyzed the contribution of each design element by comparing ablation variants that incrementally applied the structural changes proposed in this work. The compared variants included the original YOLOv11 (baseline); a variant with an added P6 layer composed of C3K2 (YOLOv11 + P6(C3K2), Variant 1); a variant with a P6 layer constructed using the pro-

posed C5Go (YOLOv11 + P6(C5Go), Variant 2); and a variant wherein the P5 layer used the proposed C5Go and the P6 layer was composed of C3K2 (YOLOv11 P5(C5Go) + P6(C3K2), Variant 3). For additional comparisons, we also evaluate a Transformer-based detector, RT-DETR [34], on the same dataset. For RT-DETR, we report the results under two training settings: (i) the same hyperparameter configuration used for YOLOv11 to provide a controlled reference and (ii) a tuned configuration tailored to RT-DETR. Furthermore, to provide an external dataset check, we additionally evaluate YOLOv11 models on a public fire/smoke dataset, FASDD-CV. The results are reported in Table 5. All experiments described in this section were conducted on the custom fire dataset.

Table 5. Comparison of performance.

Model	mAP@50	mAP@50-95	GFLOPs
YOLOv11 (baseline)	0.87	0.632	21.3
YOLOv11 + P6(C3K2) (Variant 1)	0.902	0.653	22.3
YOLOv11 + P6(C5Go) (Variant 2)	0.89	0.655	21.8
YOLOv11 P5(C5Go) + P6(C3K2) (Variant 3)	0.906	0.655	22.5
RT-DETR (YOLO hyperparams)	0.641	0.407	103.4
RT-DETR (tune for RT-DETR)	0.895	0.648	103.4
YOLOv11 (baseline, FASDD-CV)	0.291	0.127	21.3
YOLOv11 (Variant 3, FASDD-CV)	0.314	0.141	22.5

The experimental results, as presented in Table 5, indicate that the proposed variants exhibit improved performance over the YOLOv11 baseline on the custom fire dataset. Moreover, a comparison of the models' complexities in terms of GFLOPs shows that these gains are achieved with only modest computational overhead. Therefore, the proposed model is compact and particularly well suited for early fire detection—the original objective of this study.

In addition, we analyze the computational complexity using GFLOPs at the same input resolution. As shown in Table 5, the proposed variants marginally increase the GFLOPs from 21.3 to 22.5, i.e., +5.6%, while improving the detection accuracy. This indicates that the performance gains are achieved with minimal computational overhead. Compared to RT-DETR, our best variant requires 4.6 times fewer FLOPs, i.e., 22.5 vs. 103.4 GFLOPs, while achieving comparable accuracy on our dataset. This implies that the designed model is adequate for real-time early fire detection. We also performed an external dataset check on the public FASDD-CV. While the absolute accuracy differs due to dataset/domain changes, the evaluation confirms that YOLO-based detectors remain operational under the same inference process and provides an additional reference point beyond the custom dataset.

To verify the impact of the additional P6 feature level, we performed an ablation study by training and evaluating YOLOv11 under identical settings with and without P6. Table 6 reports the size-based AP, i.e., APS/APM/APL. As expected, adding P6 mainly benefits large objects, improving the APL from 0.663 to 0.675, while also yielding consistent gains in the overall mAP@50 and mAP@50-95.

Table 6. Ablation test on P6.

Model	mAP@50	mAP@50-95	APS	APM	APL
YOLOv11 baseline	0.870	0.632	0.022	0.211	0.663
YOLOv11 w/P6	0.889	0.643	0.050	0.218	0.675

In addition, we conducted qualitative experiments using real images, as shown in Figure 7. The figure presents the inference results obtained with the proposed model.

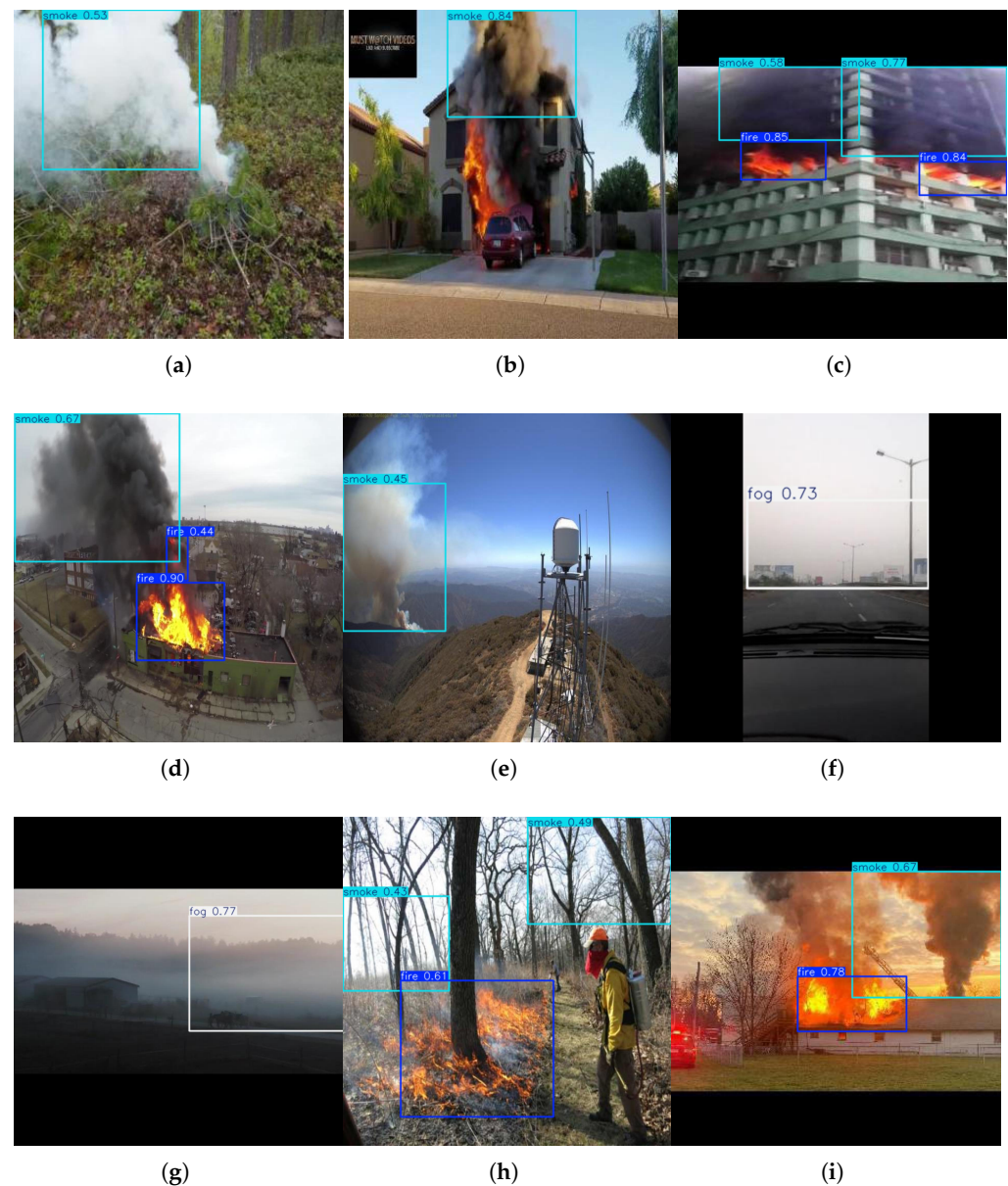


Figure 7. Qualitative test results of the proposed model in various environments: (a) Small-scale smoke rising from a wooden block; (b) Scene of a house fire; (c) Scene of a building fire; (d) Urban building fire; (e) Distant wildfire smoke viewed from a high observation tower; (f) Fog on a road; (g) Fog in an outdoor area; (h) Fire scene in a forest with a firefighter; (i) Urban fire suppression scene. Note: The boxes represent the predicted bounding boxes of the corresponding classes, and the numbers indicate the confidence scores.

The experimental results indicate that the inference bounding boxes overlap. During annotation, we adopted a non-overlapping bounding box policy whenever possible to reduce ambiguity at object boundaries and to maintain consistent labeling across annotators. This policy was intended to improve label clarity rather than to impose any constraint on the model outputs.

4. Conclusions

To reduce the false detection rate caused by the common confusion between smoke and fog in real-world scenarios, a new dataset and model architecture have been proposed. To mitigate this confusion, a novel dataset was designed that excludes overlaps in ground-truth bounding boxes, and a model was developed that enables efficient computation while enhancing performance. Training and validation experiments demonstrated a 12.86% improvement in performance, and real-world image tests confirmed the model's effectiveness.

Funding: This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP)-Innovative Human Resource Development for Local Intellectualization program grant funded by the Korea government (MSIT) (IITP-2026-RS-2023-00260267).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data supporting the findings of this study are available within the article. Additional details are available from the corresponding author upon reasonable request.

Conflicts of Interest: The author declares no conflicts of interest. The funder had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Statistics Korea. Fire Occurrence Statistics in Korea. Statistics Korea Official Website. Available online: https://www.mois.go.kr/frt/bbs/type010/commonSelectBoardArticle.do?bbsId=BBSMSTR_000000000008&nttId=100111 (accessed on 9 January 2026).
2. Chowdhury, N.; Mushfiq, D.R.; Chowdhury, A.E. Computer Vision and Smoke Sensor-Based Fire Detection System. In Proceedings of the 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 3–5 May 2019; pp. 794–798. [CrossRef]
3. Naing, Y.; Thein, T.L.L. Forest Fire Detection and Warning System for Disaster Prevention. In Proceedings of the 2023 IEEE Conference on Computer Applications (ICCA), Yangon, Myanmar, 27–28 February 2023; pp. 123–129. [CrossRef]
4. Qiao, L.; Li, S.; Zhang, Y.; Yan, J. Early Wildfire Detection and Distance Estimation Using Aerial Visible-Infrared Images. *IEEE Trans. Ind. Electron.* **2024**, *71*, 16695–16705. [CrossRef]
5. Hu, X.; Liu, W.; Wen, H.; Yuen, K.V.; Jin, T.; Junior, A.C.N.; Zhong, P. AF-Net: An Active Fire Detection Model Using Improved Object-Contextual Representations on Unbalanced UAV Datasets. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 13558–13569. [CrossRef]
6. Qadir, Z.; Le, K.; Bao, V.N.Q.; Tam, V.W.Y. Deep Learning-Based Intelligent Post-Bushfire Detection Using UAVs. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 5000605. [CrossRef]
7. Huang, Y.; Li, X.; Zhang, Z.; Wu, Q. Forest Monitoring System for Early Fire Detection Based on Convolutional Neural Network and UAV Imagery. *Sensors* **2021**, *21*, 3319. [CrossRef]
8. Goel, A.; Bhende, M.; Lal, M.; Pathak, A.; Tamrakar, P.; Thorat, P.; Sharma, S. An Emerging Fire Detection System Based on Convolutional Neural Network and Aerial-Based Forest Fire Identification. In Proceedings of the IEEE International Conference on Computer Vision and Machine Intelligence (CVMI), Allahabad, India, 10–11 December 2023; pp. 105–110. [CrossRef]
9. Toulouse, T.; Rossi, L.; Campana, A.; Celik, T.; Akhloufi, M.A. Computer vision for wildfire research: An evolving image dataset for processing and analysis. *Fire Saf. J.* **2017**, *92*, 188–194. [CrossRef]
10. Wang, Y.; Han, Y.; Tang, Z.; Wang, P. A Fast Video Fire Detection of Irregular Burning Feature in Fire-Flame Using in Indoor Fire Sensing Robots. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 7505414. [CrossRef]
11. Zhou, Y.C.; Hu, Z.Z.; Yan, K.X.; Lin, J.R. Deep Learning-Based Instance Segmentation for Indoor Fire Load Recognition. *IEEE Access* **2021**, *9*, 148771–148780. [CrossRef]
12. Cao, Y.; Tang, Q.; Wu, X.; Lu, X. EFFNet: Enhanced Feature Foreground Network for Video Smoke Source Prediction and Detection. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 1820–1832. [CrossRef]
13. Ding, J.; Yi, Y.; Wang, T.; Tian, T. Fire Video Recognition Based on Local and Global Adaptive Enhancement. *Algorithms* **2025**, *18*, 8. [CrossRef]
14. Lee, H.G.; Pham, T.N.; Nguyen, V.H.; Kwon, K.R.; Lee, J.H.; Huh, J.H. Image-Based Outlet Fire Causing Classification Using CNN-Based Deep Learning Models. *IEEE Access* **2024**, *12*, 135104–135120. [CrossRef]

15. Chaoxia, C.; Shang, W.; Zhang, F. Information-Guided Flame Detection Based on Faster R-CNN. *IEEE Access* **2020**, *8*, 58923–58930. [\[CrossRef\]](#)
16. Bothraa, R.; Temkar, R.; Ahire, D.; Mane, M.; Bhadange, R.; Jamdhade, S. Design and Development of Real-Time Image-Based Fire Detection Using OpenCV and HSV. In Proceedings of the 2024 International Conference on Advances in Computing Research on Science Engineering and Technology (ACROSET), Mumbai, India, 27–28 September 2024; pp. 1–7. [\[CrossRef\]](#)
17. Shah, R.; Gajjar, R. Real-time Fire Detection Using Image and Video Processing. In Proceedings of the 2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT), Ahmedabad, India, 29–31 August 2024; pp. 15–21. [\[CrossRef\]](#)
18. Xiong, W. Research on Fire Detection and Image Information Processing System Based on Image Processing. In Proceedings of the 2020 International Conference on Advance in Ambient Computing and Intelligence (ICAACI), Jingzhou, China, 12–13 September 2020; pp. 106–109. [\[CrossRef\]](#)
19. Chen, K.; Cheng, Y.; Bai, H.; Mou, C.; Zhang, Y. Research on Image Fire Detection Based on Support Vector Machine. *IEEE Access* **2024**, *12*, 145001–145013. [\[CrossRef\]](#)
20. Nguyen, M.D.; Vu, H.N.; Pham, D.C.; Choi, B.; Ro, S. Multistage Real-Time Fire Detection Using Convolutional Neural Networks and Long Short-Term Memory Networks. *IEEE Access* **2021**, *9*, 146667–146679. [\[CrossRef\]](#)
21. Hongyu, H.; Kuang, P.; Li, F.; Shi, H. An Improved Multi-Scale Fire Detection Method Based on Convolutional Neural Network. In Proceedings of the 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 18–20 December 2020; pp. 109–112. [\[CrossRef\]](#)
22. Deng, L.; Zhou, J.; Liu, Q. Improving YOLOv5s Algorithm for Detecting Flame and Smoke. *IEEE Access* **2024**, *12*, 126568–126575. [\[CrossRef\]](#)
23. Wang, X.; Yan, P.; Ni, R.; Li, S.; Ma, X.; Wu, Y. Fire Detection Algorithm Based on Improved YOLOv5s. In Proceedings of the 2024 9th International Conference on Intelligent Computing and Signal Processing (ICSP), Jinan, China, 19–21 April 2024; pp. 456–462. [\[CrossRef\]](#)
24. Wang, W.; Huang, Q.; Liu, H.; Jia, Y.; Chen, Q. Forest Fire Detection Method Based on Deep Learning. In Proceedings of the 2022 International Conference on Cyber-Physical Social Intelligence (ICCSI), Beijing, China, 18–21 November 2022; pp. 23–28. [\[CrossRef\]](#)
25. Li, W.; Wu, J. A YOLOv7 Forest Fire Detection System with Edge Computing. In Proceedings of the 2023 13th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 14–16 July 2023; pp. 223–227. [\[CrossRef\]](#)
26. Shahid, M.; Virtusio, J.J.; Wu, Y.H.; Chen, Y.Y.; Tanveer, M.; Muhammad, K.; Hua, K.L. Spatio-Temporal Self-Attention Network for Fire Detection and Segmentation in Video Surveillance. *IEEE Access* **2022**, *10*, 1259–1275. [\[CrossRef\]](#)
27. Ge, R.; Hu, X.; Xiao, D. Fire Detection Method Based on Improved DETR. In Proceedings of the 2024 4th International Symposium on AI (ISAI), Hangzhou, China, 25–27 April 2024; pp. 16–20. [\[CrossRef\]](#)
28. Wu, S.; Zhang, X.; Liu, R.; Li, B. A dataset for fire and smoke object detection. *Multimed. Tools Appl.* **2023**, *82*, 6707–6726. [\[CrossRef\]](#)
29. de Venâncio, 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 Comput. Appl.* **2022**, *34*, 15349–15368. [\[CrossRef\]](#)
30. Wang, M.; Yue, P.; Jiang, L.; Yu, D.; Tuo, T.; Li, J. An Open Flame and Smoke Detection Dataset for Deep Learning in Remote Sensing Based Fire Detection. *Geo-Spat. Inf. Sci.* **2024**, *28*, 511–526. [\[CrossRef\]](#)
31. Chino, D.Y.T.; Avalhais, L.P.S.; Rodrigues Jr., J.F.; Traina, A.J.M. 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.
32. Roboflow Universe: Computer Vision Datasets and Models. Online Platform. Available online: <https://universe.roboflow.com/> (accessed on 8 February 2026).
33. Ultralytics. YOLOv11—Ultralytics Documentation. Ultralytics Documentation. 2024. Available online: <https://www.ultralytics.com/> (accessed on 3 February 2026).
34. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs Beat YOLOs on Real-time Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974.

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.