

Article

SFGI-YOLO: A Multi-Scale Detection Method for Early Forest Fire Smoke Using an Extended Receptive Field

Yueming Jiang, Xianglei Meng * and Jian Wang

School of Computer and Control Engineering, Northeast Forestry University, Heilongjiang 150036, China; yueming07050106@163.com (Y.J.); ljyylm1989@163.com (J.W.)

* Correspondence: jym_nefu@nefu.edu.cn

Abstract

Forest fires pose a significant threat to human life and property. The early detection of smoke and flames can significantly reduce the damage caused by forest fires to human society. This article presents an SFGI-YOLO model based on YOLO11n, which demonstrates outstanding advantages in detecting forest fires and smoke, particularly in the context of early fire monitoring. The main principles of the algorithm include the following: first, a small-object detection head P2 is added to better extract shallow feature information; a Feature Enhancement Module (FEM) is utilized to increase feature richness, expand the receptive field, and enhance detection capabilities for small objects across multiple scales; the lightweight GhostConv is employed to significantly reduce computational costs and decrease the number of parameters; and Inception DWConv is combined with a C3k2 module to utilize multiple parallel branches, thereby enlarging the receptive field. The improved algorithm achieved a mean Average Precision (mAP50) of 95.4% on a custom forest fire dataset, surpassing the YOLO11n model by 1.8%. This model offers more accurate detection of forest fires, reducing both missed detections and false positives and thereby meeting the high precision and real-time detection requirements in forest fire monitoring.

Keywords: YOLO11; deep learning; early fire; multi-scale; small object



Academic Editor: Chunying Ren

Received: 17 July 2025

Revised: 6 August 2025

Accepted: 14 August 2025

Published: 18 August 2025

Citation: Jiang, Y.; Meng, X.; Wang, J. SFGI-YOLO: A Multi-Scale Detection Method for Early Forest Fire Smoke Using an Extended Receptive Field. *Forests* **2025**, *16*, 1345. <https://doi.org/10.3390/f16081345>

Copyright: © 2025 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

Forest fires are sudden, destructive natural disasters that are extremely difficult to contain and extinguish [1]. The catastrophic Daxing'an Mountain range fire of 6 May 1987 consumed a total area of 17,000 km² (including portions across the border), devastated 1.01 million hectares of forest within China, claimed 211 lives, injured 266 people, displaced more than 10,000 households, and left over 50,000 residents homeless. Direct economic losses exceeded RMB 500 million, with indirect losses reaching RMB 6.913 billion. On 7 January 2025, fueled by the “Santa Ana Winds,” wildfires erupted across southern California, becoming the most destructive natural disaster in U.S. history. Preliminary estimates place the damage and economic toll between USD 250 and 275 billion. Forest fires have long posed an enormous threat to forest resources and human lives. In their earliest stages, the source of a fire is typically small and easy to overlook; however, when missed, it can spread with alarming speed. The rapid and accurate detection of nascent fires, followed by immediate countermeasures, can drastically reduce losses. Therefore, developing a fast and effective fire detection system is imperative. Timely early-warning capability at the incipient stage of a fire is key to minimizing its overall impact.

Every year, thousands of forest fires erupt worldwide, causing disasters that are both immeasurable and indescribable [2]. Traditionally, detection of these blazes has long depended on ground patrols, watchtowers, sensor networks, and satellite remote sensing [3]. Owing to limitations in detection performance, economic cost, and practical operability, these conventional methods often fail to predict fires effectively and may miss fire disasters. They also struggle to meet the dual demands of high monitoring precision and complete spatial coverage over vast forested areas.

Most traditional forest fire detection algorithms based on image processing rely on hand-crafted features, such as color, motion, and texture, to delineate flame regions. Sousa J V R D et al. [4] present a real-time UAV-based system that detects forest fires in RGB and YCbCr color spaces, coupling this with an intuitive geolocation module to pinpoint fire coordinates. S. Zhong et al. [5] introduce Wi-Fire, a device-free detection framework that leverages Channel State Information (CSI) from commercial Wi-Fi equipment, using RF signal fluctuations across existing wireless infrastructure to sense fire events. Zhao J et al. [6] build a Gaussian-mixture model to segment candidate flame regions within single images, before analyzing temporal variations in color, texture, roundness, area, and contour; these statistics are combined with wavelet-based flicker frequency extracted from flame-boundary Fourier descriptors. Chino D Y T et al. [7] propose a still-image fire detection scheme that fuses color-based classification with the texture analysis of super-pixel regions to improve accuracy.

However, these shallow features often do not sufficiently characterize complex forest fire scenes, making feature extraction challenging. In recent years, with the rapid development of computer vision technology, forest fire detection has seen better solutions for complex backgrounds. The advancement of computer vision technology has provided the necessary conditions for forest fire detection, leading to deep learning-based fire detection methods gradually becoming a research focus. Currently, object detection algorithms applied in the field of fire detection include SSD [8], R-CNN [9], Faster R-CNN [10], and the YOLO series algorithms.

Li L [11] introduced PDAM-STPNet, which is a forest smoke detection network that leverages a Parallel Dual-Attention Mechanism (PDAM) to encode both local and global textures of symmetric smoke plumes, and a Small-scale Transformer Feature Pyramid Network (STPN) to markedly boost the model's capacity to spot tiny smoke objects. Li R et al. [12] proposed a high-precision, edge-focused smoke detection network featuring the SMWE module and Guillotine Feature Pyramid Network (GFPN), which enhances anti-interference capability and mitigates missed detections. L Cao. [13], based on the improved YOLO v5 model, added a plug-and-play global attention mechanism, designed a re-parameterized convolution module, and used a decoupling detection head to accelerate convergence speed. A weighted bidirectional feature pyramid network (BiFPN) [14] is introduced to merge the feature information for local information processing. According to its evaluation, the fully intersected joint (CIoU) loss function is used to optimize the multi-task loss of different types of forest fires. Ma Y et al. [15] devised a hybrid receptive-field extraction module by integrating a 2D selective scanning mechanism with residual multi-branch structures. They also introduced a dynamic-enhanced downsampling module and a scale-weighted fusion module, replacing SiLU with Mish activation to better capture flame boundaries and faint smoke textures. Soundararajan J. et al. [16] combined DeepLabV3+ with an EfficientNet-B08 backbone in a deep learning framework that uses satellite imagery to address deforestation and wildfire detection. Through advanced multi-scale feature extraction and group normalization, the system delivers robust semantic segmentation even under challenging atmospheric conditions and complex forest structures.

Currently, the YOLO series demonstrates significant advantages in the field of real-time object detection, but it is somewhat inferior to two-stage detection regarding small-object detection. Additionally, early-stage forest fire smoke and flames are relatively small and easily obscured by vegetation, making them difficult to identify, resulting in possible missed and false detections. The differences in scale between fire and smoke in images are substantial; thus, the improvement process should also consider multi-scale detection issues. Given that the deployment device is a fixed-edge device, this study proposes an early forest fire detection algorithm based on the improved SFGI-YOLO algorithm to address the aforementioned challenges. Compared to YOLO11, SFGI-YOLO incorporates the following improvements:

- To capture small targets that are overwhelmed by large-scale features in high-level feature maps, a small target detection head P2 was introduced on the shallow feature map to enhance the accuracy of detecting small targets.
- The FEM [17] introduced in place of the backbone C3k2 module enhances the richness of features through a multi-branch dilated convolution structure, extends the local perceptual capability of the network, and improves the semantic information representation for small objects, thereby enhancing the model's ability to detect small targets.
- By utilizing GhostConv [18] to reduce the number of parameters within the model, the detection efficiency of the model is enhanced.
- The Inception Depthwise Convolution (IDC) [19] is integrated into the C3k2 network structure to form the C3k2_IDC module. This is achieved by splitting large kernel convolutions into multiple parallel branches to process local and global features separately, thereby maintaining a large receptive field. This enhances the capability of detecting small objects and multi-scale features.

The remainder of this paper is structured as follows: Section 2 introduces the datasets used in this study as well as the methods and modules employed in the experiments. Section 3 presents the experimental results, ablation studies, and comparative experiments. Section 4 discusses and analyzes the model, taking into account its limitations and future work. Section 5 provides a summary of this research.

2. Materials and Methods

2.1. Dataset

The experimental dataset consisted of two parts. The first part utilizes the publicly available dataset FASDD [20], which is currently the most widely applicable and comprehensive wildfire detection dataset. The application of wildfire localization indicates that deep learning models trained on this dataset can be used to identify and monitor forest fires. These models can be deployed in watchtowers and drones, providing important references for large-scale firefighting and victim evacuation algorithm research. The second part involves collecting images from the internet, which were annotated using the labellmg software (labellmg 1.8.6) according to the FASDD labeling requirements. These annotations were reviewed and modified by team members to meet the data labeling standards. In FASDD, most scenes without flames or smoke are similar to those with flames or smoke; although they include some foggy scenes, the data quantity is slightly insufficient. Therefore, we added additional cloudy scenes for training, accounting for 5% of the dataset, to enhance the ability to distinguish between smoke and clouds. Moreover, we have introduced crown fires and some forest fire scenarios that are not included in the FASDD dataset, thereby enhancing the generalization capability of the dataset. Although this portion of images cannot fully meet the characteristics of remote sensing, by introducing flame and smoke images with a similar ratio (with fire and smoke: without fire and smoke = 7:3) to that in the FASDD dataset for this design, it partially supplements the scenes of the FASDD dataset,

thereby enhancing the generalization capability of the dataset to some extent. Figure 1 illustrates examples of images from this dataset, which is currently the most universal and comprehensive fire detection dataset.



Figure 1. Example of dataset images.

The application of wildfire localization indicates that deep learning models trained on this dataset can be used to identify and monitor forest fires. These models can be deployed simultaneously in watchtowers and drones, providing important references for algorithms concerning large-scale firefighting and victim evacuation, among other research areas. The other part consisted of images collected from the Internet. To enhance the generalization capability of the algorithm, some images were collected and annotated from the web. Although these images do not meet remote sensing characteristics, they serve as a partial supplement to the scenes in the FASDD dataset. Figure 1 provides examples of the images in this dataset.

The dataset included various scenarios such as night, day, plains, and forests. To mitigate the impact of fog, clouds, and other factors on smoke detection, the dataset includes some images of fog and clouds to assist the algorithm in differentiation. In this experiment, the dataset was randomly divided into train, validation, and test sets in a ratio of 7:2:1, resulting in a final training set of 20,776 images, a validation set of 5407 images, and a test set of 2882 images.

2.2. YOLO11

The Ultralytics team released the YOLO11 [21] algorithm in September 2024. This version inherits the advantages of YOLOv8 and improves upon it by utilizing the C3k2 module instead of the C2f module, as well as by introducing the C2PSA layer after the SPPF layer in order to enhance feature extraction capabilities, making it more suitable for applications involving small objects and complex detection tasks. YOLO11 makes significant adjustments to the model's depth and width parameters, achieving a better balance between model complexity and performance by increasing both parameters. Figure 2 illustrates the framework structure of YOLO11.

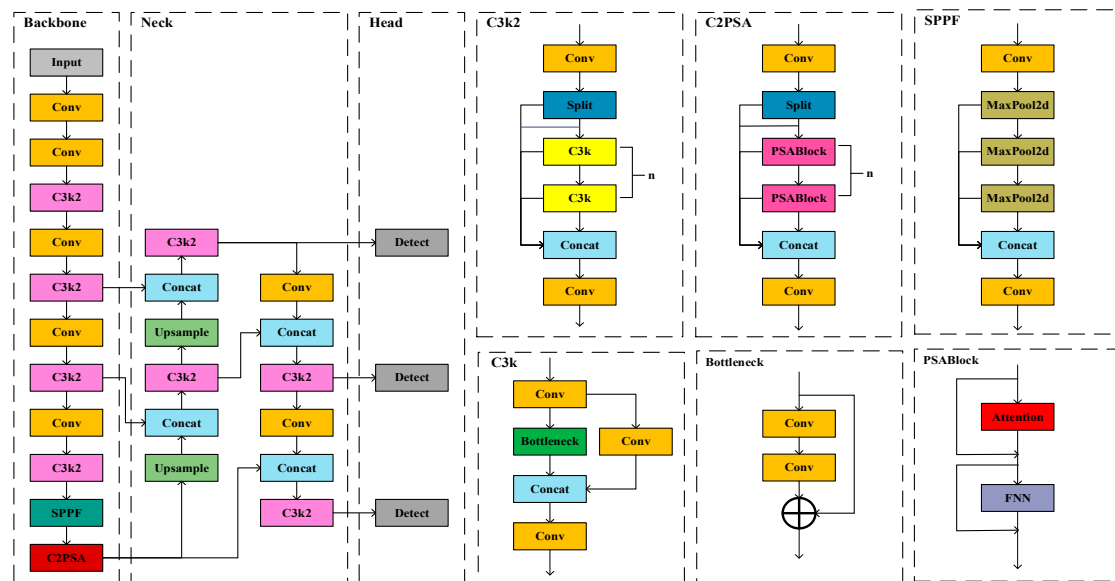


Figure 2. Structure of the YOLO11 framework.

2.3. Improved SFGI-YOLO Model

This chapter introduces the improved SFGI-YOLO model, with the framework structure being illustrated in Figure 3, where the modified parts are highlighted in red. A small target detection head P2 was added, and a Functional Enhancement Module (FEM) was introduced in the “Neck” section. The convolution layers in the framework were replaced with the lightweight GhostConv to reduce the number of parameters. Finally, the Inception DWConv was combined with C3k2 to generate the C3k2_IDC module, which maintained a large receptive field and enhanced the detection capabilities for small targets and multi-scale scenarios. This model aims to leverage the advantages of the YOLO11 framework for improvements, with the intent of developing an embedded monitoring system for forests to enhance the detection of smoke from forest fires.

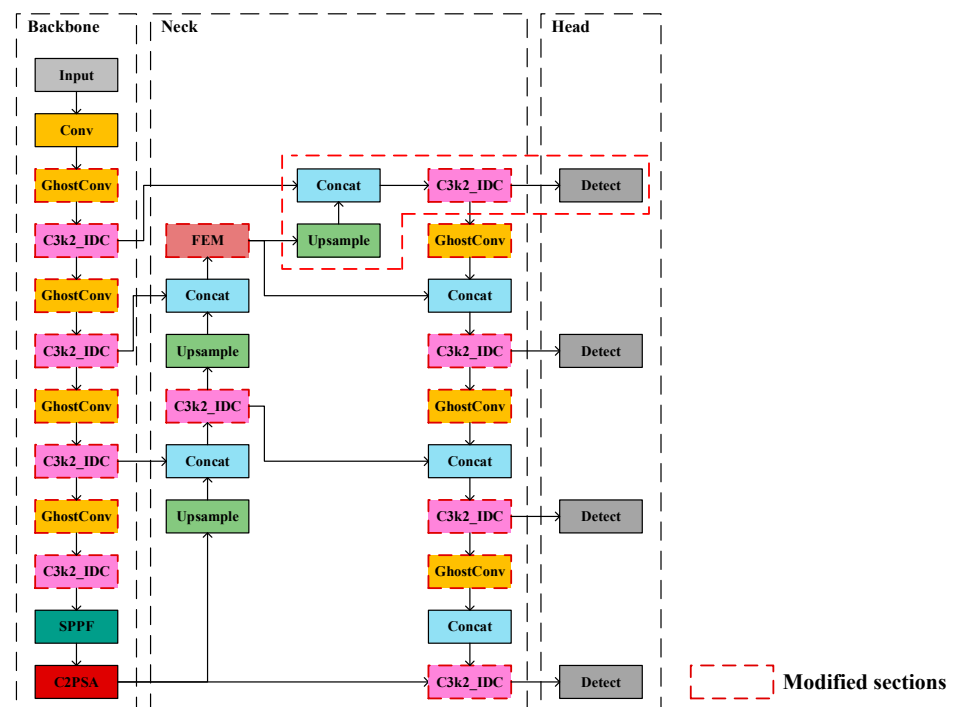


Figure 3. Improved SFGI-YOLO framework structure.

2.3.1. Small-Object Detection Head P2

YOLO11 includes three detection heads (P3, P4, and P5), each designed for detecting different sizes; their capabilities range from detecting small to large scales. However, small target objects face challenges such as insufficient features, unclear semantic representations, and the risk of feature loss due to excessive convolution. Specifically, for targets smaller than 4×4 , the scarcity of features significantly impairs the detection capability of the P3 head. Consequently, a P2 detection head has been introduced to enhance the detection performance for small targets.

YOLO11 innovatively applies decoupled detection heads, separating the classification and regression branches. The classification branch utilizes depthwise separable convolutions (DWConv) to reduce computational burden. The structural diagram of “Detect” is shown in Figure 4.

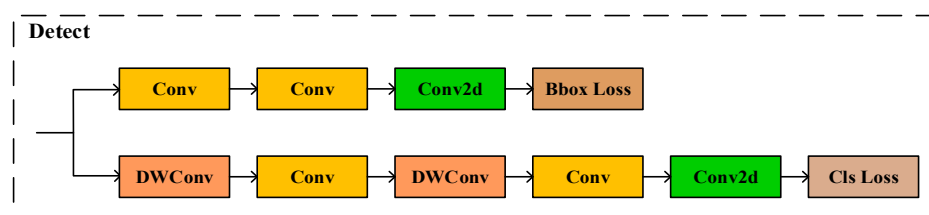


Figure 4. Detect structural diagram.

However, in small-object detection, accuracy can be adversely affected by low resolution or insufficient feature focus. Although the YOLO SAHI method can be employed for detection, this would increase detection time, making it unsuitable for applications that require high real-time performance, such as in relation to forest fire detection. Therefore, this approach introduces a small-object detection head, P2, which utilizes lower-level convolutional features to detect shallower feature maps, thereby better capturing the detailed information of small targets and avoiding the loss of such details in traditional high-level feature maps. When combined with other detection heads, the small-object detection head P2 can also help mitigate multi-scale issues to a certain extent.

2.3.2. FEM

To address the difficulty of efficiently extracting small target feature information in backbone networks, which leads to the challenge of distinguishing small targets from the background, FFCA-YOLO proposed three innovative plug-and-play modules—FEM, FFM, and SCAM. These three modules enhance the network’s local perception ability, multi-scale feature fusion capability, and global association ability across channels and spaces, respectively. Based on the requirements of this design scheme, the FEM was utilized in the algorithm to improve the detection capability for small targets.

The FEM enhances the feature representation of small targets through the following two approaches: (1) increasing feature richness by utilizing a multi-branch structure to extract more discriminative semantic information; (2) expanding the receptive field by applying dilated convolution to obtain richer local contextual information, thereby better understanding the surrounding environment of small targets.

The mathematical expression for the FEM is as follows:

$$W_1 = f_{conv}^{3 \times 3} \left[f_{conv}^{1 \times 1}(F) \right] \quad (1)$$

$$W_2 = f_{dconv}^{3 \times 3} \left\{ f_{conv}^{3 \times 1} \left\{ f_{conv}^{1 \times 3} \left[f_{conv}^{1 \times 1}(F) \right] \right\} \right\} \quad (2)$$

$$W_3 = f_{dconv}^{3 \times 3} \left\{ f_{conv}^{1 \times 3} \left\{ f_{conv}^{3 \times 1} \left[f_{conv}^{1 \times 1}(F) \right] \right\} \right\} \quad (3)$$

$$Y = \text{Cat}(W1, W2, W3) \oplus f_{conv}^{1 \times 1}(F) \quad (4)$$

Among them, $f_{conv}^{1 \times 1}$, $f_{conv}^{1 \times 3}$, $f_{conv}^{3 \times 1}$, and $f_{conv}^{3 \times 3}$ represent standard convolution operations with kernel sizes of 1×1 , 1×3 , 3×1 , and 3×3 , respectively. $f_{dconv}^{3 \times 3}$ denotes a dilated convolution operation with a dilation rate of 5. $\text{Cat}(\cdot)$ indicates the operation of concatenating feature maps. $\oplus$ denotes the element-wise addition operation of feature maps. F is the input feature map. $W1$, $W2$, $W3$ represent the output feature maps after the first three branches undergo standard convolution and dilated convolution. Y is the output feature map of the FEM.

The FEM enhances the model's learning capability through multi-branch attribute convolution, resulting in richer local contextual features, thus improving the feature representation capability for small target objects. The structural diagram of the FEM is illustrated in Figure 5.

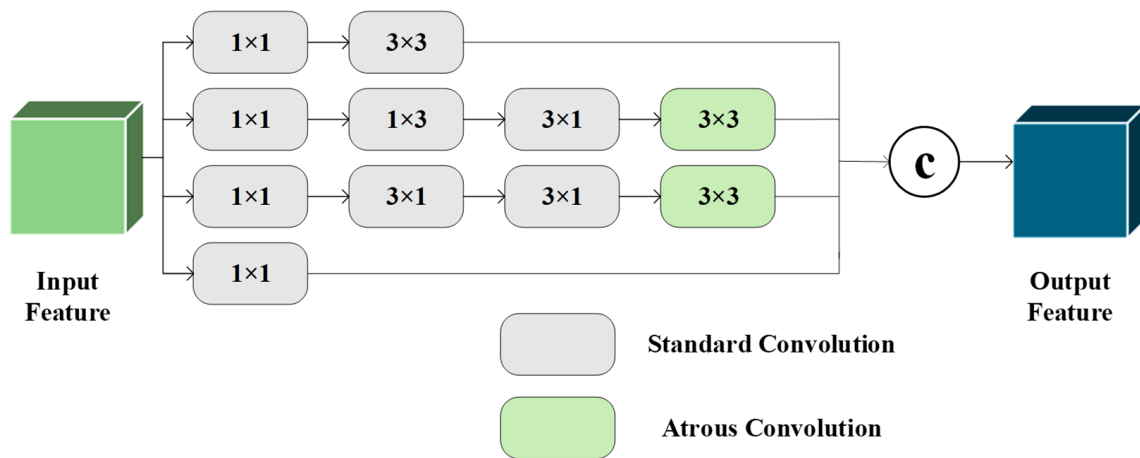


Figure 5. Structural diagram of the FEM.

The FEM consists of only two branches that incorporate dilated convolution. Each branch performs a 1×1 convolution operation on the input feature maps to preliminarily adjust the number of channels for subsequent processing. The first branch implements a residual structure that forms an equivalent mapping, preserving the critical feature information of small objects. The other branches are cascaded using convolution kernels of different sizes (1×3 , 3×3 , etc.), adding additional convolutional layers to extract more contextual information and enhance the richness of the feature maps.

In the detection of small target flames and smoke, the FEM captures rich local features through a multi-branch dilated convolution structure, enhancing the model's ability to distinguish between the target and the background. It mitigates background confusion and improves detection capabilities in low-resolution or complex background scenarios, thereby enhancing detection accuracy and robustness.

2.3.3. GhostConv

The feature maps extracted using conventional CNN networks contain a significant amount of redundant information. However, embedded devices have limited memory and computational resources, making it difficult to deploy convolutional neural networks on such devices. Considering that the redundant information in the feature map layers might be an important component of a successful model, these redundant feature maps were not eliminated during the design of lightweight models. Instead, an attempt was made to obtain these redundant feature maps with lower computational costs. GhostConv exhibits significant differences from the Conv operation, as illustrated in Figure 6.

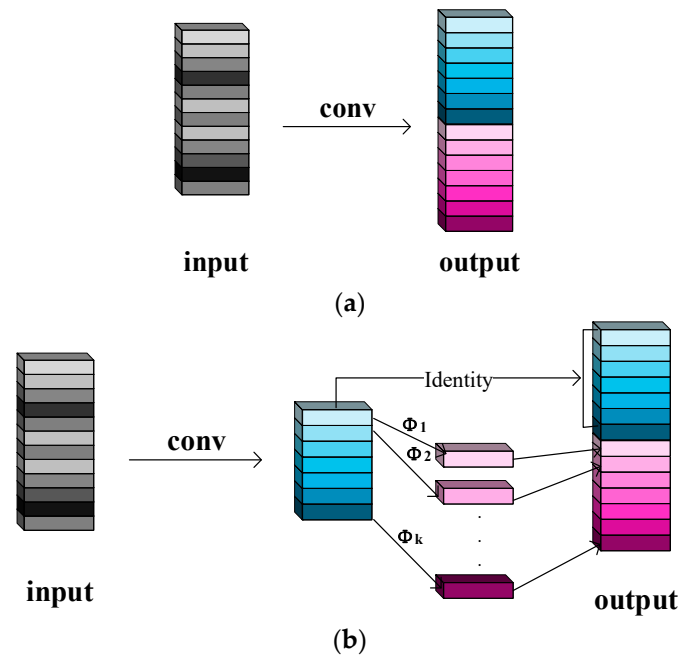


Figure 6. Conv and GhostConv operational procedure. (a) Conv operational procedure. (b) GhostConv operational procedure.

- Conv widely employs 1×1 point convolution units, while the main convolution in GhostConv can have a customizable convolution kernel size.
- Conv uses point convolution to process features across channels, followed by depthwise convolution to handle spatial information. In contrast, GhostConv first generates a few intrinsic feature maps and then employs inexpensive linear operations to enhance features and increase the number of channels.
- The operations for handling each feature map in Conv are limited to depthwise convolution or shifting operations in previous efficient architectures, whereas the linear operations in GhostConv can exhibit significant diversity.

GhostConv is divided into three steps—conventional convolution, Ghost generation, and feature map concatenation. Figure (b) illustrates the operational process of GhostConv. Firstly, a conventional convolution is performed to generate a feature map with a smaller number of channels, thus reducing computational load. Then, based on the obtained feature map, depthwise convolution is applied, which is a cost-effective operation that further reduces computational costs and generates a new feature map. Finally, the two sets of feature maps are concatenated to produce the final output.

The formula for conventional convolution calculation is as follows:

$$Y = X * f + b \quad (5)$$

$*$ denotes the convolution operation; b is the bias unit; $Y \in R^{h' \times \omega' \times n}$ is the output feature map; h' is the output height; ω' is the output width; and n is the output dimension, corresponding to the number of filters. $f \in R^{c \times k \times k \times n}$ represents the convolutional filters; c is the number of channels; k is the height and width of the filters; n is the output dimension. The FLOPs during the convolution operation are calculated as $n \times h' \times \omega' \times c \times k \times k$, where n and c are often very large during computation.

The formula for GhostConv is as follows:

$$Y' = X * f' \quad (6)$$

$Y' \in R^{h' \times \omega' \times m}$; m is the number of kernels in the initial convolution. $f' \in R^{c \times k \times k \times m}$, with the remaining content being the same as for Conv. After the GhostConv operation, the FLOPs are generally reduced to $\frac{1}{s}$ of the original, where s is the number of Ghost feature maps produced.

Thus, when GhostConv is incorporated into the algorithm, the number of parameters and computational costs are significantly reduced, e.g., the computational time was reduced from 0.118937 s to 0.108418 s. This leads to a decrease in GPU usage during the forest fire detection process, minimizing latency and facilitating the rapid detection of smoke and flames.

2.3.4. C3k3_IDC

Inception DWConv reduces the high computational cost associated with directly using large-kernel convolutions by decomposing the large-kernel DWConv into four parallel branches. These four parallel branches include a small kernel convolution, two orthogonal strip convolution kernels, and an identity mapping.

The first branch uses a 3×3 small-kernel convolution to process local information, as defined in Equation (7), which retains detailed information while saving computational costs. The second and third branches use 1×11 and 11×1 convolutions, as described in Equations (8) and (9), respectively, effectively capturing global information in terms of width and height. The fourth branch employs identity mapping, bypassing convolution operations, as outlined in Equation (10), thus reducing unnecessary computations and enhancing computational efficiency.

$$X'_{hw} = DWconv_{3 \times 3}^{g \rightarrow g}(X_{hw}) \quad (7)$$

$$X'_w = DWconv_{1 \times 11}^{g \rightarrow g}(X_w) \quad (8)$$

$$X'_h = DWconv_{11 \times 1}^{g \rightarrow g}(X_h) \quad (9)$$

$$X'_{id} = X_{id} \quad (10)$$

Finally, the mathematical formula is obtained by stitching together the outputs of each branch as follows:

$$X' = Concat(X'_{hw}, X'_w, X'_h, X'_{id}) \quad (11)$$

By processing features at different scales, the receptive field of the network can be expanded. However, the importance of features at different scales may not be uniform in certain scenarios. After feature fusion, there may be issues such as high-frequency noise or discontinuities at boundaries. Utilizing concatenate operations to further process the fused features can alleviate aliasing effects and enhance the expressive power of the features.

SFGI-YOLO combines C3k2 with the Inception DWConv module to maintain a large receptive field, enhancing its detection capabilities for small objects and multi-scale targets, this is conducive to extracting information in complex backgrounds; therefore, in nighttime scenarios, Inception DWConv enhances the detection capabilities of YOLO11n for fires and smoke. The structures of the Inception DWConv module and C3k2_IDC module are shown in Figure 7. The C3k2_IDC module effectively improves accuracy and recall in fire detection processes, reducing the probabilities of missed and false detections, making it suitable for application in relation to forest fire detection.

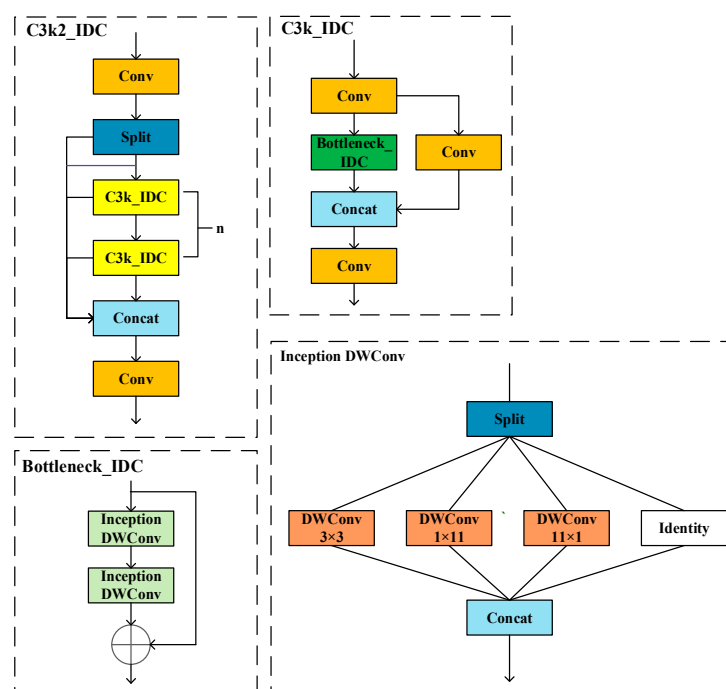


Figure 7. C3k2_IDC overall structural diagram.

3. Results

3.1. Experimental Environment

The configuration of the software and hardware in the experimental environment is presented in Table 1, while the settings for the algorithm training parameters are provided in Table 2.

Table 1. Hardware and software configuration.

Configuration	Version
GPU	RTX 4090D × 1 card
CPU	18 cores
CUDA	12.4
Pytorch	2.5.1
Python	3.12

Table 2. Algorithm parameter configuration.

Parameter	Configuration
epochs	200
batch	16
imgsz	640 × 640
lr0	0.01
workers	8
conf	0.3
IoU	0.5

3.2. Evaluation Criteria

To evaluate the performance of the algorithm, this design adopts the following parameters as evaluation metrics: Precision, Recall, mean Average Precision at an IoU threshold of 0.5 (mAP@0.5), mean Average Precision at IoU thresholds of 0.5 to 0.95 (mAP@0.5–0.95), parameters, GFLOPs, and FPS.

In calculating Precision, Recall, and mAP, the following parameters are utilized: True Negative (TN, hereinafter referred to as TN), which correctly predicts non-smoke and non-flames; True Positive (TP, hereinafter referred to as TP), which correctly predicts smoke and flames; False Negative (FN, hereinafter referred to as FN), which incorrectly predicts smoke and flames as non-smoke and non-flames; and False Positive (FP, hereinafter referred to as FP), which incorrectly predicts non-smoke and non-flames as smoke and flames.

Precision indicates the proportion of correctly predicted samples among those predicted as smoke and flames. The calculation formula is as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (12)$$

Recall represents the proportion of samples of smoke and flames that were correctly predicted as smoke and flames among all samples of smoke and flames. The calculation formula is as follows.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (13)$$

To calculate mAP, it is necessary to first compute the Average Precision (AP). AP integrates Precision and Recall for a comprehensive assessment, and mAP is the mean of the AP across different categories. The formulas for AP and mAP are as follows:

$$AP = \int_0^1 (\text{Precision})d(\text{Recall}) \quad (14)$$

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i \quad (15)$$

where n represents the number of categories and, in this design, the categories are as follows: 0—fire; 1—smoke, $n = 2$. AP_i is the average precision for each category i .

FPS represents the frames per second, which effectively reflects the computational speed of the algorithm and is used to evaluate the real-time performance of the algorithm. The algorithm components in this experiment were all conducted on the same GPU with a consistent utilization rate. “Parameters” indicates the number of parameters, and “GFLOPs” signifies the floating point operations per second, which are often used to assess the computational cost of the model; hence, parameters and GFLOPs should be as small as possible when applied to embedded devices.

In this experiment, the confidence interval during the algorithm training process was set between 0 and 1, while during the experiment, to reduce the probabilities of missed detections and false detections, the confidence interval was set to above 0.3.

During the algorithm training process, it was observed that excessive training discussions during the selection of epochs could lead to overfitting. Consequently, after testing, it was ultimately decided to select 200 epochs. Additionally, an early stopping directive was implemented during the training, halting the process when mAP@50–95 did not show improvement after 100 epochs, with mAP@50–95 being chosen as the best value for the final round.

3.3. Ablation Test

To explore the enhancement of SFGI-YOLO’s performance, six groups of ablation experiments were conducted; the results are shown in Table 3. This experiment aimed to assess the feasibility and effectiveness of the improved module.

Table 3. Ablation test.

	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6	Exp 7	Exp 8
Detection heads P2		✓				✓	✓	✓
FEM			✓			✓	✓	✓
GhostConv				✓			✓	✓
C3k2_IDC					✓			✓
Precision (B)	91.8	92.6	92.7	92.6	93.2	93.0	92.9	93.6
Recall (B)	90.7	91.6	90.8	90.2	90.5	91.7	91.9	92.4
mAP50 (B)	94.0	94.9	94.2	93.8	94.2	95.3	95.4	95.4
mAP50–95 (B)	75.8	76.4	76.4	75.9	75.8	77.2	77.4	77.6
AP50 _{fire} (B)	88.6	90.4	88.9	88.1	88.9	91.1	91.4	91.4
AP50 _{smoke} (B)	99.4	99.5	99.5	99.5	99.5	99.5	99.5	99.5
Parameters (M)	2.6	2.7	3.4	2.3	2.9	2.9	2.6	2.6
GFLOPs	6.3	10.2	13.6	5.5	7.6	14.0	13.1	14.5
FPS	322.6	294.1	285.7	312.5	312.5	270.3	270.3	294.1

“✓” indicates the modules added in this Exp.

In Exp 1, it was observed that the parameters of YOLO11 demonstrated a good performance. Following the introduction of new modules, the performance was further enhanced.

Comparisons revealed that the inclusion of detection head P2 and the FEM led to improvements in Precision by 0.8 and 0.9, Recall by 0.9 and 0.1, mAP50 by 0.9 and 0.2, and mAP50–95 by 0.6 for both, respectively. However, this resulted in an increase in parameters and GFLOPs, with parameters rising by 0.1 and 0.8, GFLOPs increasing by 3.9 and 7.3, and a decrease in FPS by 28.5 and 36.5. This demonstrates that while the introduction of detection head P2 and the FEM increases the computational load and parameter count, they significantly enhance the model’s ability to detect flames and smoke, effectively reducing the chances of missed detections and false alarms.

In Exp 3, when replacing the model’s Conv with GhostConv, the model’s Precision improved by 0.8, Recall decreased by 0.5, mAP50 dropped by 0.2, mAP50–95 increased by 0.1, parameters decreased by 0.3, GFLOPs reduced by 0.8, and FPS dropped by 10.1. It is evident that GhostConv is advantageous for reducing the parameter and computational load of the module, with minimal impact on other evaluation metrics, thus mitigating the increase in parameters and computation caused by the introduction of additional modules.

When the C3k2 module was replaced with C3k2_IDC, a comparison with Exp 1 showed that Precision increased by 1.4, Recall decreased by 0.2, mAP50 rose by 0.2, mAP50–95 remained unchanged, parameters increased by 0.3, GFLOPs increased by 1.3, and FPS decreased by 10.1. The C3k2_IDC module processes local and global features through four branches, maintaining a large receptive field and enhancing the detection capability for small objects and multiple scales, effectively improving the detection ability for flames and smoke across different scales.

Through the comprehensive comparison of Exp 1, Exp 6, Exp 7, and Exp 8, it was found that the AP50_{smoke} initially achieved a high precision level of 99.4%, which improved by 0.1% as the modules were added sequentially. Meanwhile, AP50_{fire} increased from 88.6% to 91.4%, and other metrics also showed improvement. This indicates that the model’s detection capability for specific elements such as fire and smoke has been enhanced.

Ultimately, SFGI-YOLO achieved an increase in Precision by 1.8, Recall by 1.7, mAP50 by 1.4, and mAP50–95 by 1.8, while keeping parameters unchanged; GFLOPs increased by 8.2 and FPS decreased by 28.5. Therefore, it is evident that although the introduction of new modules leads to an increased computational load and a decline in FPS, the overall performance remains strong, demonstrating high accuracy in fire and smoke detection, with significant performance enhancements that meet real-time detection requirements.

3.4. Comparative Experiments and Analysis

In this study, targets with pixels smaller than 32×32 were classified as small targets, targets with pixels greater than 32×32 but less than 96×96 were classified as medium targets, and targets with pixels greater than 96×96 were classified as large targets.

The training dataset consisted of 20,776 images, of which 11,172 contained flames and smoke, as well as 9604 comparative images. There were 45,095 detection targets in this dataset, which includes 8780 small targets, 12,748 medium targets, and 23,567 large targets.

The validation dataset contained 5407 images, with 3014 images containing flames and smoke, as well as 2393 comparative images. There were 5410 detection targets in this dataset, which includes 2415 small targets, 3457 medium targets, and 5783 large targets.

The comparative images in the training and validation sets include cloud interference images used to enhance the ability to distinguish between clouds and smoke, images without flames and smoke from the same scene, and other forest images from different scenes that enrich the dataset.

3.4.1. Comparative Experiment

Comparative analysis shows that SFGI-YOLO exhibits an increase of 1.8% in Precision, 1.7% in Recall, 1.4% in mAP50, and 1.8% in mAP50–95 compared to YOLO11n, indicating significant improvements.

To comprehensively evaluate and validate the detection capabilities of SFGI-YOLO, this section will detail comparative experiments with YOLOv9n, YOLOv10n, YOLO11n, YOLOv12n, RTDETR50, etc. The experiments will compare performance based on Precision, Recall, mAP50, mAP50–95, parameters, GFLOPs, FPS, and other metrics.

The comparison results are presented in Table 4.

Table 4. Comparison experiment.

	YOLOv9n	YOLOv10n	YOLO11n	YOLOv12n	RTDETR	SFGI-YOLO
Precision (B)	92.7	91.2	91.8	92.5	92.4	93.6
Recall (B)	90.9	90.2	90.7	90.3	91.1	92.4
mAP50 (B)	94.1	93.9	94.0	94.0	93.7	95.4
mAP50–95 (B)	76.1	75.2	75.8	75.6	72.8	77.6
Parameters (M)	2.0	2.7	2.6	2.6	41.9	2.8
GFLOPs	7.6	8.2	6.3	6.3	125.6	14.5
FPS	303.0	344.8	322.6	277.8	107.5	263.2

From the above comparative experiments, it can be observed that SFGI-YOLO demonstrates superior performance in terms of accuracy compared with other conventional models. Although there was a slight decrease in FPS, it met real-time requirements. SFGI-YOLO effectively balances factors such as detection accuracy, detection speed, and model complexity.

3.4.2. Image Comparison Experiment

To validate the performance of SFGI-YOLO and YOLO11n in practical scenarios, a comparison was conducted under various conditions, including small targets, long distances, and multiple scales. The results are demonstrated through visualized images.

As shown in Figure 8, under cloud interference, YOLO11 failed to detect one instance of smoke, while the confidence levels of the smoke detected by SFGI-YOLO were all higher than those of YOLO11. SFGI-YOLO effectively reduced the interference of clouds in smoke detection, as well as demonstrating the robustness of this algorithm, as the presence of clouds did not affect the ability to detect smoke.

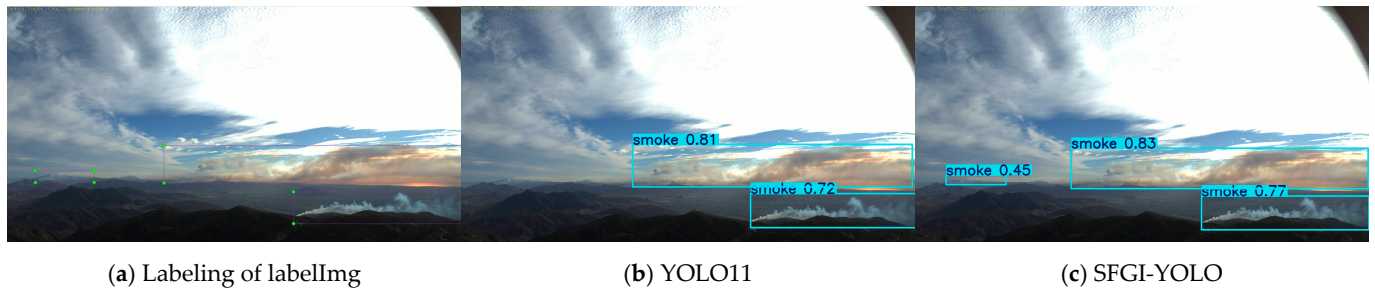


Figure 8. Results of forest fire and smoke detection under cloudy conditions.

Small-target detection is of the utmost importance in forest fire detection. As shown in Figures 9 and 10, the confidence of SFGI-YOLO in small-target detection was significantly higher than that of YOLO11. In Figure 9, both YOLO11 and SFGI-YOLO detected flames, but the confidence level of SFGI-YOLO was higher than that of YOLO11. Under nighttime conditions, flames were not detected by YOLO11, whereas SFGI-YOLO demonstrated a higher confidence level.

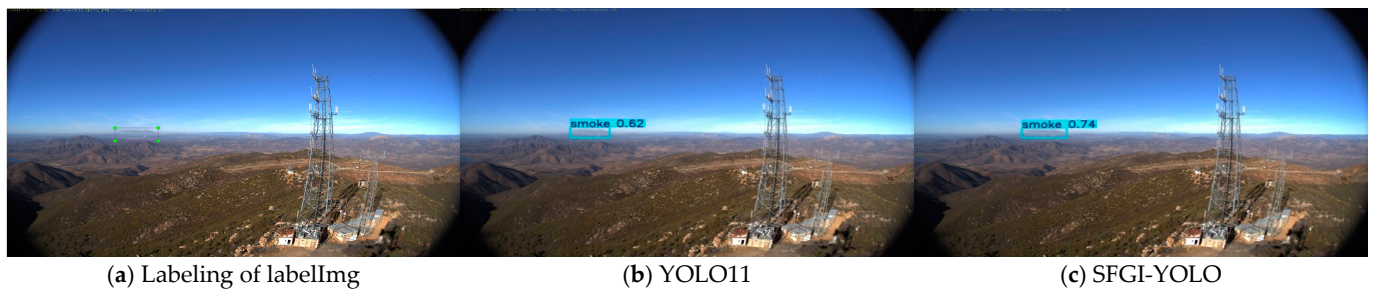


Figure 9. Results of forest fire and smoke detection under conditions of distant small targets.



Figure 10. Results of forest fire and smoke detection under low visibility conditions at night.

In multi-scale and occlusion scenarios, as shown in Figures 11 and 12, YOLO11's ability to distinguish smoke boundaries and detect flames obscured by occlusions is slightly inferior to that of SGFI-YOLO. Moreover, the confidence level of the detected flames and smoke in SGFI-YOLO is also higher than that of YOLO11.

It can be seen that in various scenarios such as small targets, long distances, and multiple scales, SFGI-YOLO outperforms YOLO11 in terms of detection capabilities at smoke boundaries and obscured flames, as well as in relation to the overall detection accuracy.

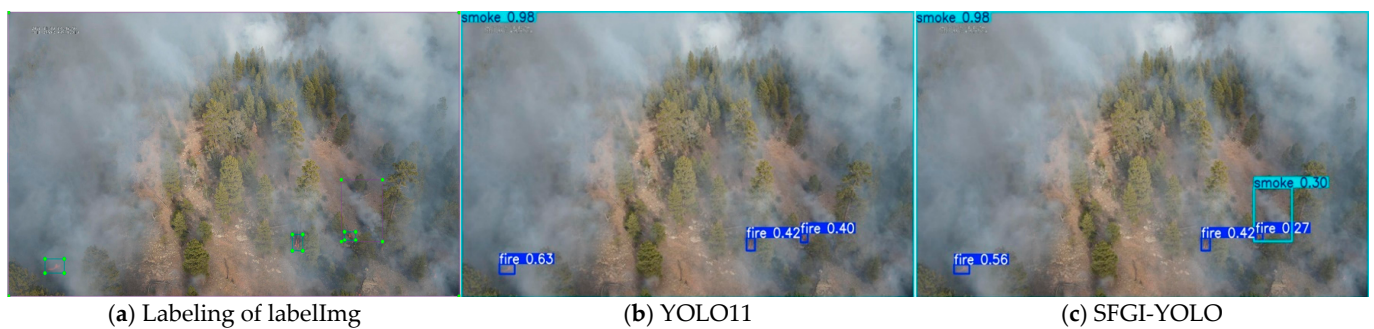


Figure 11. Results of forest fire and smoke detection under multi-scale conditions.

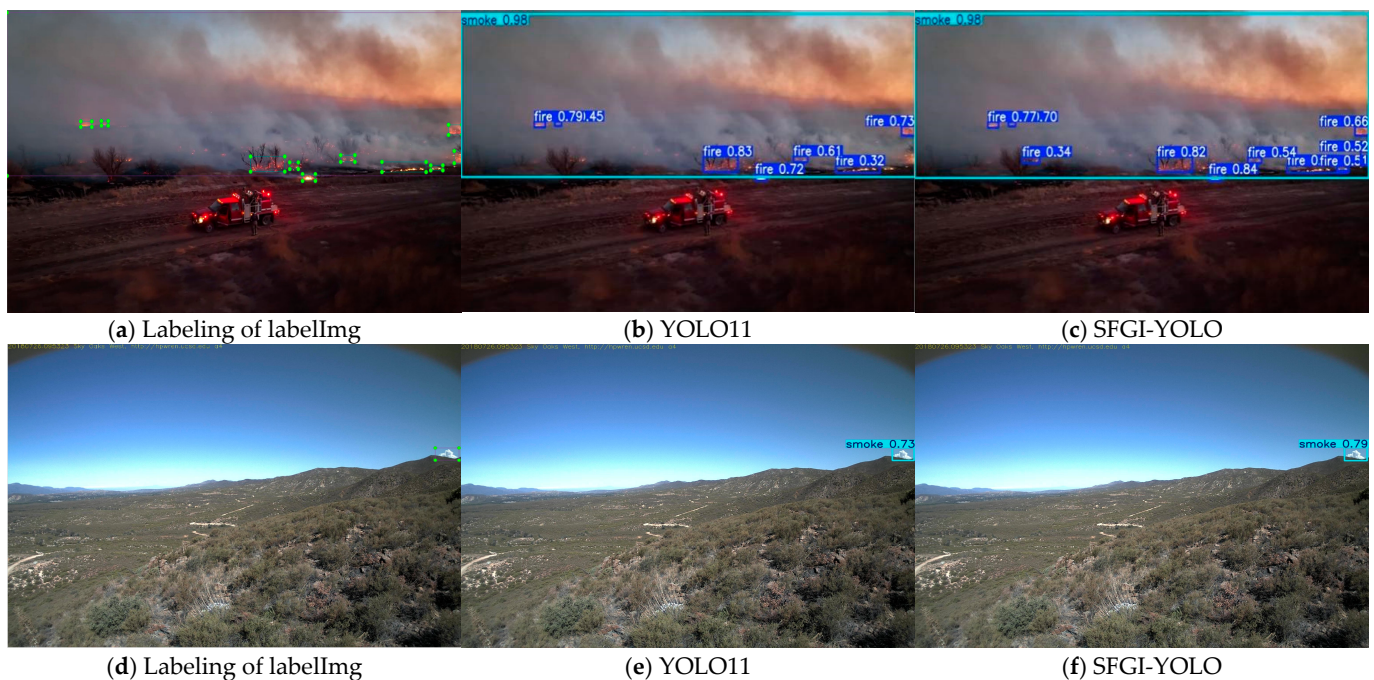


Figure 12. Results of forest fires and smoke under obscured conditions.

4. Discussion

At present, the issue of forest fires remains serious globally. The frequency of fires caused by climatic and human factors is continually increasing. These fires not only destroy vast areas of forest and damage the habitats of fauna and flora, but also pose threats to human safety and property. Therefore, the ability to detect and identify forest fires in a timely and accurate manner is crucial for effectively controlling the spread of fires.

Deep neural networks, particularly object detection algorithms, are utilized to automatically learn and identify the characteristics of flames and smoke, analyzing complex patterns in image and video data to achieve the rapid and precise detection of fire scenarios. The current object detection algorithms used in the field of fire detection include SSD, Faster R-CNN, and YOLO series algorithms, among others. Despite YOLO11, a single-stage algorithm, having a lower accuracy compared to two-stage detection systems, it offers significant advantages in real-time performance, making it more suitable for forest fire detection.

In this study, improvements were made to YOLOv11 by adding a small-object detection head (P2) to detect shallower feature maps, addressing the multi-scale issue in conjunction with other detection heads. A Feature Enhancement Module (FEM) that utilizes a multi-branch structure is employed to extract more discriminative semantic information, thereby

increasing feature richness. Additionally, dilated convolutions are applied to obtain richer local contextual information, expanding the receptive field and enhancing the capability to detect small objects across multiple scales. A lightweight GhostConv is used to generate a portion of intrinsic feature maps with a small number of standard convolutions, followed by inexpensive linear operations to produce additional Ghost features. Ultimately, the intrinsic feature maps are concatenated with the Ghost feature maps, resulting in a number of feature maps that are equivalent to traditional convolutions while reducing computational costs and the number of parameters. By combining Inception DWConv with the C3k2 module and utilizing multiple parallel branches, the receptive field is further increased. The improved algorithm achieves a high accuracy and fast frames per second, meeting the requirements for high-precision and real-time detection in forest fire monitoring.

Considering environmental factors, the design of response plans for natural disasters is outlined as follows. Humid air can cause metal rust and circuit aging in electronic modules. Thus, when electronic devices are exposed to clouds and fog for extended periods, it is essential to implement certain moisture-proof measures during the initial production phase. One approach could be to “block the invasion paths” by selecting active protective measures such as waterproof enclosures and moisture-proof boxes, supplemented by passive measures such as desiccants. The regular replacement of carriers is necessary to prevent accidents such as collapse or falling. In the event of natural disasters such as landslides or floods, drones should be deployed to inspect the damaged equipment, facilitating monitoring within a short time frame. Once conditions stabilize, a new site can be selected to construct a new detection network.

However, this model still has room for improvement. Although the parameter count of this model is comparable to YOLO11n, the GFLOPs exhibited a certain enhancement. Therefore, there remains potential for progress regarding the accuracy, speed, and model size of parallel models. Environmental factors, such as strong winds, dense fog, and night-time conditions, can affect the detection of fires and smoke. Hence, it is essential to consider how to gather data and expand the dataset in order to enhance the model’s processing capabilities. Additionally, this model is expected to be deployed in embedded modules to build monitoring systems. Consequently, it is important to consider the distribution of cameras based on forest conditions and ensure timely alerts in cases of network instability as part of future research.

5. Conclusions

This design is based on an improved YOLO11 algorithm, the SFGI-YOLO model, which aims to overcome the issues in relation to flame and smoke detection in forest fires that are associated with previously used methods. The model introduces a small target detection head (P2) to extract shallower feature information and utilizes a Feature Enhancement Module (FEM) to enhance the representation of small target features. Additionally, Conv layers in the algorithm are replaced with GhostConv to reduce parameters and lower computational costs. Ultimately, the IDC module is combined with the C3k2 module to create the C3k2_IDC module, which processes local and global features in a parallel branch manner, while maintaining a large receptive field. This enhances the capabilities of detecting small targets and multi-scale objects.

The SFGI-YOLO model achieves a precision of 93.6%, a recall of 92.4%, an mAP50 of 95.4%, and an mAP50–95 of 77.6% on the forest fire dataset used in this design. The model has 2.8 million parameters, 14.5 GFLOPs, and an FPS of 263.2. Although it is slightly slower than YOLO11, it demonstrates superior accuracy and performance, making it more suitable for deployment in embedded devices. Future work will involve expanding the dataset to include various fire scenarios in order to further improve the model’s detection accuracy.

Author Contributions: Conceptualization: X.M. and Y.J.; methodology: Y.J.; software: X.M.; validation: X.M. and Y.J.; formal analysis: X.M.; investigation: Y.J.; resources: J.W.; data curation: Y.J.; writing—original draft preparation: X.M.; writing—review and editing: Y.J.; visualization: X.M.; supervision: J.W.; project administration: J.W.; funding acquisition: Y.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by Central university projects (2572022BH03).

Data Availability Statement: The data and the conclusions of this study are available from the corresponding author upon reasonable request.

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

Abbreviations

The following abbreviations are used in this manuscript:

P2,P3,P4,P5	Detection heads of different scales
FEM	Feature Enhancement Module
lr0	Initial learning rate
conf	Confidence threshold
IoU	Intersection of Union
IDC	Inception Depthwise Convolution
EXP	Experiment

References

- Chen, Y.; Li, J.; Sun, K.; Zhang, Y. A lightweight early forest fire and smoke detection method. *J. Supercomput.* **2024**, *80*, 9870–9893. [\[CrossRef\]](#)
- Alkhatib, A. A Review on Forest Fire Detection Techniques. *Int. J. Distrib. Sens. Netw.* **2014**, *2014*, 103–110. [\[CrossRef\]](#)
- Li, X.; Zhang, G.; Tan, S.; Yang, Z.; Wu, X. Forest Fire Smoke Detection Research Based on the Random Forest Algorithm and Sub-Pixel Mapping Method. *Forests* **2023**, *14*, 485. [\[CrossRef\]](#)
- Sousa, J.V.R.D.; Gamboa, P.V. Aerial Forest Fire Detection and Monitoring Using a Small UAV. *KnE Eng.* **2020**, *5*, 242–256. [\[CrossRef\]](#)
- Zhong, S.; Huang, Y.; Ruby, R.; Wang, L.; Qiu, Y.-X.; Wu, K. Wi-fire: Device-free fire detection using WiFi networks. In Proceedings of the IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6.
- Zhao, J.; Zhang, Z.; Han, S.; Qu, C.; Yuan, Z.; Zhang, D. SVM based forest fire detection using static and dynamic features. *Computer Sci. Inf. Syst.* **2011**, *8*, 821–841. [\[CrossRef\]](#)
- Chino, D.Y.T.; Avalhais, L.P.S.; Rodrigues, 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; pp. 95–102.
- Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. *arXiv* **2016**, arXiv:1512.02325. [\[CrossRef\]](#)
- Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *arXiv* **2014**, arXiv:1311.2524. [\[CrossRef\]](#)
- Girshick, R. Fast R-CNN. *arXiv* **2015**, arXiv:1504.08083. [\[CrossRef\]](#)
- Zhan, J.; Hu, Y.; Cai, W.; Zhou, G.; Li, L. PDAM-STPNNNet: A Small Target Detection Approach for Wildland Fire Smoke through Remote Sensing Images. *Symmetry* **2021**, *13*, 2260. [\[CrossRef\]](#)
- Li, R.; Hu, Y.; Li, L.; Guan, R.; Yang, R.; Zhan, J.; Cai, W.; Wang, Y.; Xu, H.; Li, L. SMWE-GFPNNNet: A high-precision and robust method for forest fire smoke detection. *Knowl.-Based Syst.* **2024**, *289*, 111528. [\[CrossRef\]](#)
- Cao, L.; Shen, Z.; Xu, S. Efficient forest fire detection based on an improved YOLO model. *Vis. Intell.* **2024**, *2*, 20. [\[CrossRef\]](#)
- Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; IEEE: New York, NY, USA, 2020.
- Ma, Y.; Shan, W.; Sui, Y.; Wang, M.; Wang, M. LHRF-YOLO: A Lightweight Model with Hybrid Receptive Field for Forest Fire Detection. *Forests* **2025**, *16*, 1095. [\[CrossRef\]](#)
- Soundararajan, J.; Kalukin, A.; Malof, J.; Xu, D. Deep Learning-Driven Multi-Temporal Detection: Leveraging DeeplabV3+/Efficientnet-B08 Semantic Segmentation for Deforestation and Forest Fire Detection. *Remote Sens.* **2025**, *17*, 2333. [\[CrossRef\]](#)

17. Zhang, Y.; Ye, M.; Zhu, G.; Liu, Y.; Guo, P.; Yan, J. FFCA-YOLO for Small Object Detection in Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5611215. [[CrossRef](#)]
18. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features From Cheap Operations. In Proceedings of the 2020 IEEE CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; IEEE: New York, NY, USA, 2020.
19. Yu, W.; Zhou, P.; Yan, S.; Wang, X. InceptionNeXt: When Inception Meets ConvNeXt. *arXiv* **2023**, arXiv:2303.16900.
20. Wang, M.; Jiang, L.; Yue, P.; Yu, D.; Tuo, T. FASDD: An Open-access 100,000-level Flame and Smoke Detection Dataset for Deep Learning in Fire Detection. *Earth Syst. Sci. Data Discuss.* **2022**, preprint.
21. Glenn, J.; Jing, Q. Ultralytics YOLO11. 2024. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 15 June 2025).

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.