scientific reports

Check for updates

OPEN

Research and application of deep learning object detection methods for forest fire smoke recognition

Luhao He[1,2,3], Yongzhang Zhou[1,2,3]✉, Lei Liu[1,2,3], Yuqing Zhang[1,2,3] & Jianhua Ma[1,2,3]✉

Forest fires are severe ecological disasters worldwide that cause extensive ecological destruction and economic losses while threatening biodiversity and human safety. With the escalation of climate change, the frequency and intensity of forest fires are increasing annually, underscoring the urgent need for effective monitoring and early warning systems. This study investigates the application effectiveness of deep learning-based object detection technology in forest fire smoke recognition by using the YOLOv11x algorithm to develop an efficient fire detection model. The objective is to enhance early fire detection capabilities and mitigate potential damage. To improve the model's applicability and generalizability, two publicly available fire image datasets, WD (Wildfire Dataset) and FFS (Forest Fire Smoke), encompassing various complex scenarios and external conditions, were employed. After 501 training epochs, the model's detection performance was comprehensively evaluated via multiple metrics, including precision, recall, and mean average precision (mAP50 and mAP50-95). The results demonstrate that YOLOv11x excels in bounding box loss (box loss), classification loss (cls loss), and distribution focal loss (dfl loss), indicating effective optimization of object detection performance across multiple dimensions. Specifically, the model achieved a precision of 0.949, a recall of 0.850, an mAP50 of 0.901, and an mAP50-95 of 0.786, highlighting its high detection accuracy and stability. Analysis of the precision–recall (PR) curve revealed an average mAP@0.5 of 0.901, further confirming the effectiveness of YOLOv11x in fire smoke detection. Notably, the mAP@0.5 for the smoke category reached 0.962, whereas for the flame category, it was 0.841, indicating superior performance in smoke detection compared with flame detection. This disparity primarily arises from the distinct visual characteristics of flames and smoke; flames possess more vivid colors and defined shapes, facilitating easier recognition by the model, whereas smoke exhibits more ambiguous and variable textures and shapes, increasing detection difficulty. In the test set, 86.89% of the samples had confidence scores exceeding 0.85, further validating the model's reliability. In summary, the YOLOv11x algorithm demonstrates excellent performance and broad application potential in forest fire smoke recognition, providing robust technical support for early fire warning systems and offering valuable insights for the design of intelligent monitoring systems in related fields.

Keywords Deep learning, Object detection, YOLOv11x, Forest fire, Flame detection, Smoke recognition, Early warning

The increasing frequency of forest fires worldwide poses severe threats to ecosystems, climate stability, and human safety. Over the past decade, both the frequency and intensity of forest fires have risen markedly. Statistics indicate that from 2002 to 2016, more than 420 million hectares of forests were destroyed annually by fires worldwide, resulting in substantial ecological and economic losses[1,2]. The intensification of climate change and human activities has further increased the frequency of forest fires, exerting immense pressure on ecosystem recovery[3]. Traditional fire monitoring methods, such as manual inspections, fire danger index models, and ground-based sensors, struggle to promptly detect and locate fires in complex terrains and variable climatic conditions because of their slow response times and limited accuracy, thereby posing significant challenges to early warning systems and effective firefighting efforts[4].

In recent years, deep learning has demonstrated remarkable advantages in image processing and pattern recognition, offering innovative solutions to address the timeliness and complexity of forest fire monitoring.

[1]Center for Earth Environment and Earth Resources, Sun Yat-sen University, Zhuhai 519000, Guangdong, China. [2]School of Earth Sciences and Engineering, Sun Yat-sen University, Zhuhai 519000, Guangdong, China. [3]Key Laboratory of Geological Processes and Mineral Resources Exploration, Zhuhai 519000, Guangdong Province, China. ✉email: zhouyz@mail.sysu.edu.cn; 05161935@cumt.edu.cn

Unlike traditional methods, deep learning can efficiently process large volumes of remote sensing data and video surveillance images, significantly enhancing the accuracy of fire smoke detection, particularly in rapidly spreading fire scenarios[5,6]. Therefore, exploring the application of deep learning in forest fire detection not only improves monitoring timeliness and accuracy but also provides new approaches for strengthening fire emergency response capabilities.

Three main methods are employed for deep learning-based forest fire detection: image classification, image segmentation, and object detection[7]. Image classification uses convolutional neural networks (CNNs) to analyze images captured by satellites or drones, distinguishing fire-affected areas from normal forest regions. Common models include AlexNet[8], VGGNet[9], Inception (GoogleNet)[10], DenseNet[11], and MobileNet[12]. For example, Akilandeswari et al. (2024)[13] compared the performance of SqueezeNet and VGG in fire detection and reported that SqueezeNet is suitable for resource-constrained devices, whereas VGG offers higher accuracy. To enhance detection performance, Ilyas et al. (2024)[14] integrated support vector machines (SVMs) with CNNs, and Reis et al. (2023)[15]improved the model's detection capabilities by extracting features via pretrained CNN models and fine-tuning them. However, the image classification method typically relies on global features for prediction, which makes it vulnerable to background complexity and unclear fire details. As a result, the accuracy of detecting small or fragmented fire areas may decrease[15–19].

Semantic segmentation achieves more precise fire detection and analysis by classifying each pixel in an image as a fire or background, thereby enhancing monitoring detail and providing reliable information for fire impact assessments. Common models include U-Net[20], FCN[21], DeepLab[22], SegNet[23], and PSPNet[24]. For example, Guan et al. (2022)[5]developed the MaskSU R-CNN, which achieved excellent performance on the FLAME dataset, with 91.85% accuracy and 82.31% mIoU. Additionally, Shirvani et al. (2023)[25]proposed the RAUNet method, which utilized Sentinel-2 data to perform high-resolution near-real-time fire detection, and the results were outstanding (IoU = 0.9238, overall accuracy = 0.985). However, segmentation models may produce higher error rates in complex backgrounds, multi-scale variations, or dynamic environments, and their fire detail recognition may still be less flexible than object detection methods[26–28].Object detection combines image classification and segmentation by identifying target categories and providing bounding box information, enabling accurate localization of fires[29]. In fire monitoring, the timeliness and accuracy of object detection facilitate the early identification of potential fire hazards, thereby reducing fire-related losses. It leverages the strengths of both image classification and segmentation, allowing for accurate identification of fire location and category, particularly when dealing with multi-scale, small targets, and complex backgrounds[30,31]. Unlike image classification and segmentation, object detection not only provides bounding box information for fire-affected regions but also performs real-time fire localization, which is essential for rapid responses[5,32,33]. Typical object detection frameworks include the two-stage R-CNN series and the single-stage YOLO series[34–40]. Two-stage methods typically generate region proposals first and then perform classification and bounding box regression within those regions. Although these methods generally offer high accuracy, their computational intensity and slower processing speed make them less suitable for real-time applications like forest fire monitoring[41,42]. In contrast, the YOLO series is a single-stage method that divides an image into grids and performs object detection on each grid, skipping the region proposal stage. This significantly reduces computation time and model complexity, allowing YOLO to achieve faster inference in real-time systems while maintaining a low false positive rate[43]. Furthermore, YOLO's end-to-end training optimizes both object classification and localization tasks, effectively addressing small target detection and multi-scale challenges in complex environments. As a result, YOLO, with its rapid inference and efficient computational resource usage, is the preferred method for this study, particularly in monitoring forest fires with dynamic changes and complex backgrounds. YOLO provides real-time fire localization, ensuring swift responses and high accuracy[37]. Previous YOLO versions have already been widely used in fire monitoring[44–46], demonstrating good accuracy and efficiency in practical applications. However, there is still room for improvement, especially in extreme weather, dynamic conditions, and low-contrast environments, where finer fire detail recognition is needed[44,47]. YOLOv11 continues the real-time advantages of its predecessors while improving its ability to handle small targets and complex backgrounds through an enhanced feature extraction module and multi-scale training strategies[43]. This allows it to deliver higher detection accuracy and faster response times in dynamic fire monitoring scenarios. Additionally, YOLOv11's improvements in computational optimization and multi-scale feature fusion[43] ensure accurate fire localization even in complex backgrounds, effectively tackling common issues like background clutter and target scale variations in fire monitoring[47].

Thus, this study employs the YOLOv11 algorithm for intelligent forest fire monitoring. Initially, remote sensing images are annotated and augmented during the data preprocessing phase to improve model generalization. YOLOv11 is then used to detect fire smoke and flames, ensuring robustness across different weather conditions and terrain. Finally, precision, recall, and mean average precision (mAP) are used to quantitatively evaluate YOLOv11's performance in fire detection. This study aims to enhance real-time forest fire monitoring capabilities by utilizing YOLOv11, offering more accurate and efficient technical support for fire warning systems and advancing the application of intelligent monitoring technologies in disaster management.

Algorithm introduction

YOLOv11 is the latest real-time object detector released by the Ultralytics team on September 30, 2024. Building upon improvements from YOLOv9 and YOLOv10, it represents a significant advancement in real-time object detection technology[48]. YOLOv11 features an enhanced architectural design, more sophisticated feature extraction techniques, and more refined training methods, enabling it to capture complex details in images more effectively. This model extensively supports tasks such as real-time object detection, instance segmentation, and pose estimation and can accurately identify objects of various orientations, scales, and sizes[49]. Compared with earlier versions, YOLOv11 introduces several notable enhancements, including the following:

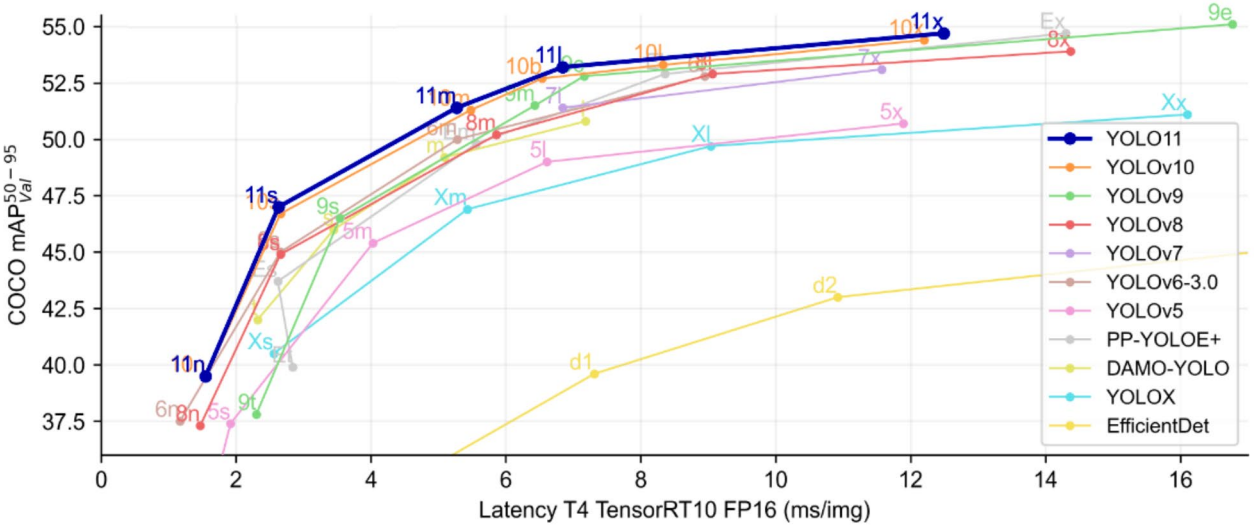

Fig. 1. **Benchmarking YOLOv11 against Previous Versions**[50].

Model	Size (pixels)	mAP$^{val}_{50-95}$	Improvement (%)	Speed$^{CPU\ ONNX}$ (ms)	Speed$^{T4\ TensorRT10}$ (ms)	params (M)	FLOPs (B)
YOLO11n	640	39.5	-	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	18.98%	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	9.57%	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11	640	53.4	3.69%	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	2.43%	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

Table 1. Performance of the five different size models of YOLOv11[50].

the improvement of backbone and neck architectures with the addition of the C3k2 and C2PSA modules, which significantly boost feature extraction capabilities and increase detection accuracy; optimized architectural design and training processes that achieve faster processing speeds while maintaining a strong balance between accuracy and performance; YOLOv11m achieves greater mean average precision (mAP) on the COCO dataset with a 22% reduction in parameters than YOLOv8m does, thereby improving computational efficiency without compromising accuracy; cross-environment adaptability allows deployment on various platforms, including edge devices, cloud platforms, and systems supporting NVIDIA GPUs; and comprehensive task support encompasses multiple computer vision applications, such as object detection, instance segmentation, image classification, pose estimation, and oriented object detection (Fig. 1)[50].

For the benchmark analyses, YOLOv11 was compared with several previous generation models, as shown in Table 1. The results demonstrate that YOLOv11 achieves an mAP50-95 of 54.7% on the COCO dataset, significantly outperforming the YOLOv5 to YOLOv10 series (with mAP50-95 values of approximately 53%, 50%, 48%, 45%, 43%, and 40%, respectively) and exhibiting faster inference speeds. Particularly in the detection of small objects and multiscale detection within complex scenes, YOLOv11 exhibits superior precision. Additionally, YOLOv11 offers multiple submodels (such as YOLOv11n, YOLOv11 s, YOLOv11m, YOLOv11l, and YOLOv11x) to meet different application requirements. With respect to inference latency, the submodels have latencies ranging from 1.5 to 12.5 milliseconds. For example, YOLOv11 s achieves approximately 47% mAP50–95 with a latency of 2.5 milliseconds, YOLOv11m reaches approximately 51% mAP50–95 with a latency of 5 milliseconds, and the highest precision model, YOLOv11x, attains approximately 55% mAP50–95 with a latency of 12 milliseconds. These results indicate that YOLOv11 successfully balances high real-time performance with accuracy. Compared with other object detection models, such as EfficientDet and DAMO-YOLO, YOLOv11 achieves higher mAPs50–95 under the same latency conditions. For example, at a latency of 7 milliseconds, EfficientDet and DAMO-YOLO achieve mAP50-95 values of approximately 40% and 50%, respectively, whereas YOLOv11l approaches 53% mAP50-95. This balance between precision and latency gives YOLOv11 a competitive edge in real-time detection scenarios that require both high accuracy and low latency.

In the forest fire early warning system, smoke detection presents several challenges: it often appears as blurred and irregular shapes with unclear boundaries; the smoke may be small in scale and unevenly distributed, making it easy to confuse with natural phenomena like clouds or fog; additionally, the forest environment is complex, with diverse backgrounds and varying lighting conditions that can affect detection performance. Therefore, selecting an appropriate object detection algorithm for this scenario requires consideration of factors such as detection accuracy (mAP), real-time performance (FPS), inference delay, computational complexity,

and adaptability to complex environments. Quantitatively, YOLOv11 stands out for its real-time performance, offering high inference speed (80–150 FPS) and low latency (7-12ms), making it ideal for rapid responses in fire warning systems (Table 2). It is well-suited for real-time detection on surveillance cameras and drones. However, its ability to detect small targets, such as distant smoke, and handle complex backgrounds is limited, which could lead to missed detections in early fire detection. In comparison, Faster R-CNN excels in small target detection and adapting to complex environments (mAP 65-70%, high complex background adaptability), making it effective for capturing blurred and unclear boundary smoke features. However, its slower inference speed (5–10 FPS) and higher latency (150-200ms) make it more suitable for high-precision, non-real-time analysis, such as offline image review or training high-precision models. EfficientDet-D4 strikes a good balance between accuracy and real-time performance (mAP 55-60%, inference speed 25–40 FPS), with a low computational complexity (50–75 GFLOPs), making it ideal for deployment on edge devices in forest monitoring systems that require both accuracy and real-time responsiveness. RetinaNet, with its use of Focal Loss to address class imbalance, achieves an mAP of 60-65%. This makes it particularly effective in forest environments with significant background interference (such as leaves and fog), though its inference speed is slower than YOLOv11 and EfficientDet. Thus, it can serve as a backup model to improve detection accuracy when combined with other algorithms. SSD offers fast inference speed (50–80 FPS) and low latency (10-20ms), but its lower accuracy (mAP 45-50%) makes it better suited for large-scale post-fire monitoring rather than early smoke detection. DETR excels in capturing global contextual features (mAP 65-70%) and recognizing small targets, such as distant smoke, especially in complex environments with overlapping targets. However, due to its longer inference delay (250-300ms), it is more suitable for high-precision offline analysis or data labeling. Cascade R-CNN, with its multi-stage detection mechanism, improves small target detection accuracy (mAP 70-75%), making it particularly useful for capturing distant or blurred smoke features. However, its slower inference speed (3–7 FPS) makes it more appropriate for use as a post-processing algorithm in high-precision tasks. CenterNet, using keypoint detection, excels at identifying irregularly shaped smoke (mAP 55-60%), and it strikes a good balance between real-time performance (25–35 FPS) and latency (70-100ms), making it suitable for real-time monitoring with drones or forest edge devices. In summary, for real-time smoke detection in forest fire scenarios, YOLOv11 is recommended as the core detection algorithm, combined with EfficientDet or RetinaNet to improve accuracy. For offline, high-precision analysis or verification of suspected fires, Faster R-CNN, Cascade R-CNN, or DETR can be employed for deeper analysis to ensure comprehensive and accurate detection. Additionally, for resource-constrained edge devices, CenterNet can provide real-time monitoring support as a compact deployment solution. This multi-model fusion strategy optimally leverages the strengths of each algorithm in terms of real-time performance, accuracy, and adaptability to complex environments, significantly improving the detection capabilities of the forest fire smoke recognition system.

Network structure

As illustrated in Fig. 2, YOLOv11's network architecture comprises three main components, namely, the backbone, neck, and head, ensuring high efficiency and accuracy in object detection tasks[49]. The backbone is responsible for extracting rich features from the input image and includes convolutional layers, C3k2 modules, spatial pyramid pooling (SPPF) modules, and channel and spatial attention (C2PSA) modules. The neck fuses multilayer features from the backbone through upsampling, feature concatenation, and C3k2 modules, thereby enhancing the model's ability to detect objects of varying scales with greater accuracy. The head converts the fused feature maps into actual detection results, providing information on the object locations, confidence scores, and class labels. Compared with previous models such as YOLOv8, YOLOv11 introduces several significant technical innovations. These include the optimization and expansion of the cross-stage partial (CSP) bottleneck architecture, allowing the model to switch between C3k2 and bottleneck blocks by flexibly setting the C3k parameters to adapt to different tasks and data characteristics. Additionally, the C2PSA mechanism incorporates a multihead attention mechanism that combines channel and spatial attention, significantly enhancing feature representation and the detection of small objects. The detect head integrates depthwise convolutions (DWConv), which substantially reduce the number of parameters and computational costs, thereby improving inference efficiency and feature extraction capabilities. This design also enhances information flow, increases detection performance in complex scenes and for small objects, and optimizes the interaction between classification and regression tasks, reducing the risk of overfitting[49]. Figures 3 and 4 depict the structural designs of the C3k2 and C2PSA modules, further highlighting YOLOv11's advancements in feature extraction and attention mechanisms.

Model	mAP (%)	FPS	Inference Latency (ms)	GFLOPs	Applicable Scenarios
YOLOv11[51]	55–60	80–150	7–12	20–30	Real-time video analysis, autonomous driving, mobile applications
Faster R-CNN[52]	65–70	5–10	150–200	200–300	Medical imaging, document analysis, satellite imagery
EfficientDet-D4[53]	55–60	25–40	50–80	50–75	Edge computing, industrial inspection, mobile devices
RetinaNet[54]	60–65	20–30	80–120	150–200	Security surveillance, small object detection, industrial applications
SSD (VGG16)[55]	45–50	50–80	10–20	50–100	Real-time traffic monitoring, drone vision, augmented reality
DETR[56]	65–70	10–15	250–300	500–600	Autonomous driving, image segmentation, complex scene detection
Cascade R-CNN[57]	70–75	3–7	200–300	350–450	Industrial quality control, medical diagnostics, satellite image analysis
CenterNet[58]	55–60	25–35	70–100	100–150	Pedestrian detection, traffic sign recognition, real-time surveillance

Table 2. Quantitative Comparison of Object Detection Algorithms.

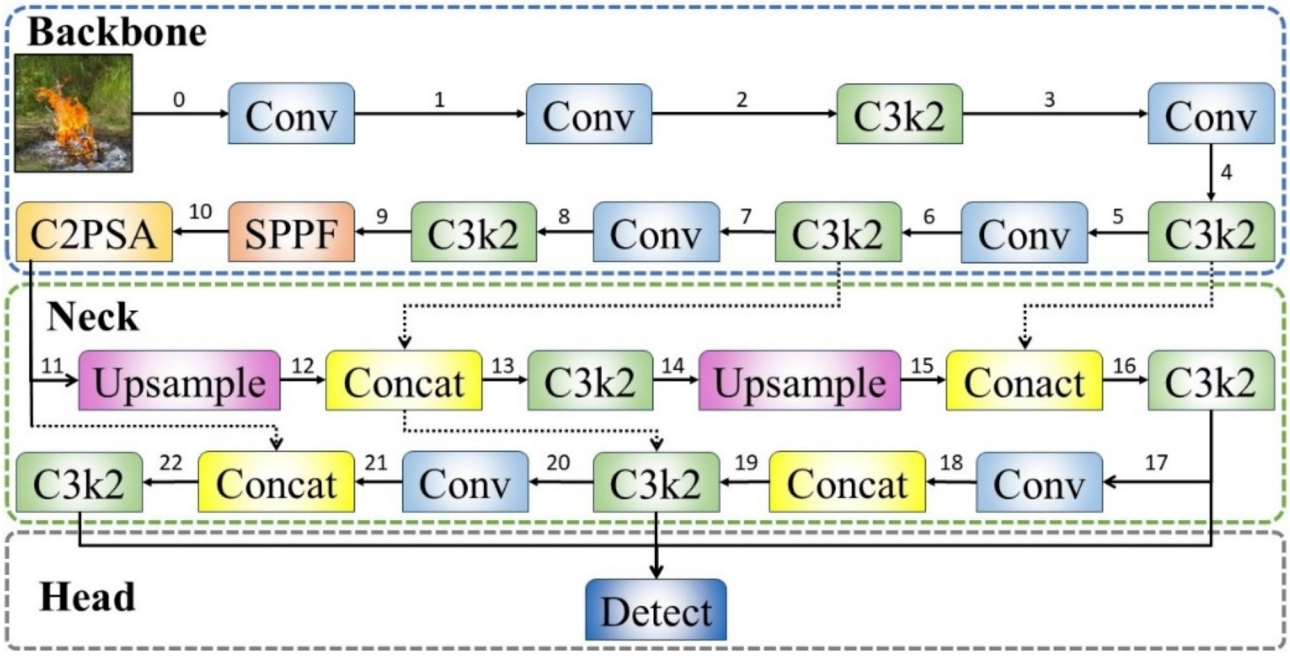

Fig. 2. YOLOv11 network structure diagram[75].

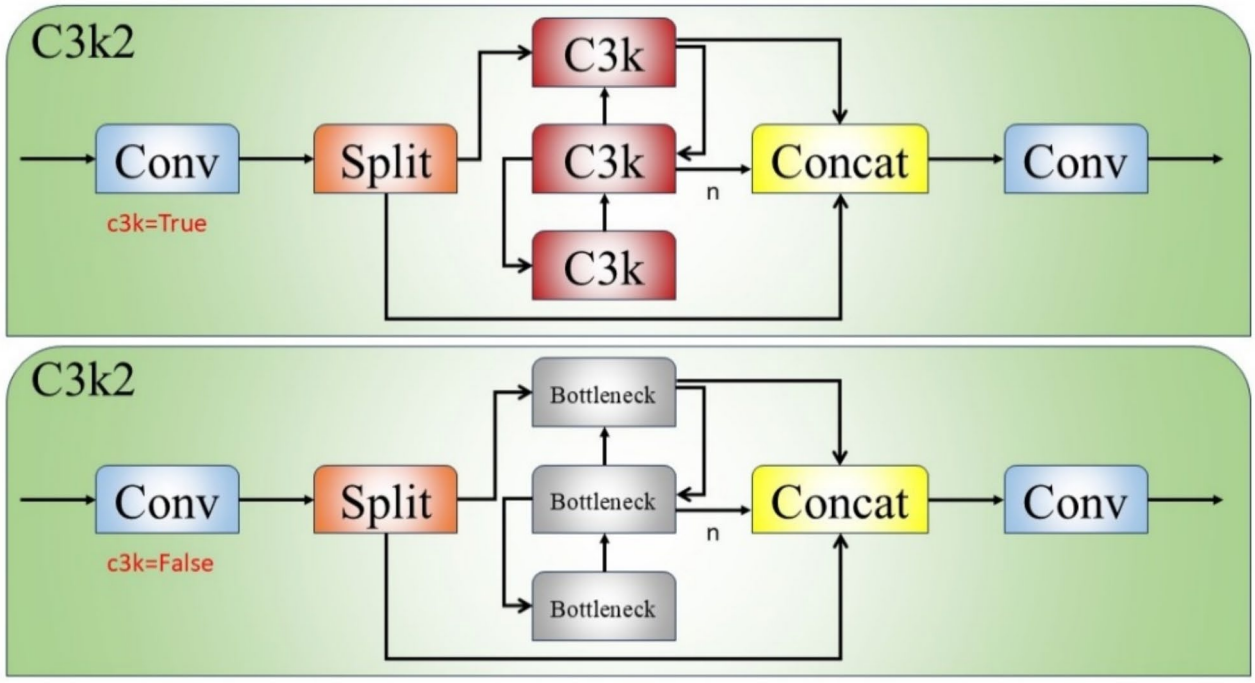

Fig. 3. **C3k2 module**[49].

YOLOv11 technical features

YOLOv11 introduces several key technical innovations in the field of object detection compared with its predecessors, such as YOLOv8, primarily in three areas. First, the C3k2 module optimizes and extends the cross-stage partial (CSP) bottleneck architecture, building on the design principles of the C2f module. By flexibly adjusting the C3k parameters, the model can alternate between C3k2 and bottleneck blocks to better suit specific tasks and data characteristics. When the C3k parameter is set to True, the model employs C3k2 blocks to capture complex features effectively, making it suitable for high-dimensional and diverse data processing. Conversely,

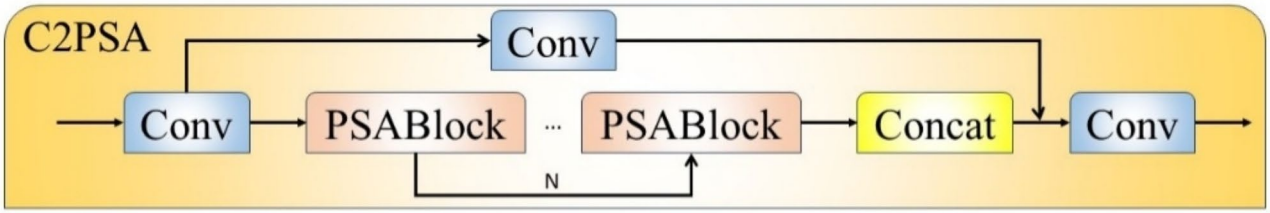

Fig. 4. **C2PSA module**[49].

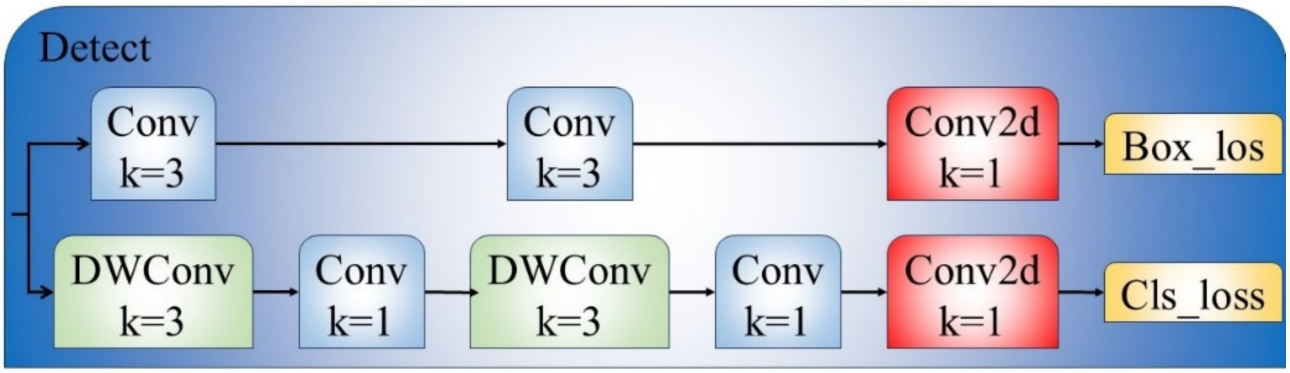

Fig. 5. **Detection module**[49].

setting the parameter to false switches the model to bottleneck blocks, which are more efficient in terms of parameter count and computational complexity. This flexibility enables the model to adapt quickly to various application requirements, thereby enhancing overall adaptability (Fig. 3).

Second, the channel and spatial attention (C2PSA) mechanism extends the original C2 mechanism by embedding a multihead attention mechanism. By combining channelwise and spatialwise attention, C2PSA significantly improves feature representation and the detection of small objects. This mechanism optimizes feature selection and reduces computational complexity. The incorporation of multihead attention allows the model to focus on multiple feature subspaces simultaneously, thereby capturing richer information and enhancing both the model's generalizability and real-time application performance. Additionally, the flexible structure of C2PSA integrates seamlessly with other network components, making it suitable for a wide range of tasks and scenarios, thus further increasing the model's adaptability and efficiency (Fig. 4)[49].

Finally, YOLOv11's decoupled head design introduces two depthwise convolutions (DWConv), which significantly reduce the number of parameters and computational costs while enhancing inference efficiency. DWConv not only strengthens feature extraction capabilities and improves information flow—thereby enhancing detection performance in complex scenes and for small objects—but also optimizes the interaction between classification and regression tasks, mitigating the risk of overfitting. These comprehensive improvements collectively increase the model's performance and stability (Fig. 5).

Dataset acquisition and processing

This study utilized two publicly available fire image datasets: WD (Wildfire Dataset)[59] and FFS (Forest Fire Smoke)[60]. These datasets provide a substantial number of real-world fire scene images, with FFS primarily focusing on forest fires and WD encompassing a diverse range of wildfire conditions across different environments. The inclusion of urban and rural fire instances further enhances the dataset's representativeness. Additionally, these datasets incorporate a variety of weather conditions (e.g., clear, foggy, rainy), lighting scenarios (e.g., daytime, nighttime), and terrain types (e.g., mountainous, flatlands), ensuring a robust and diverse training foundation for fire detection models. As illustrated in Fig. 6, the datasets primarily include two detection targets: fire and smoke. Specifically, the FFS dataset consists of 42,900 images, evenly distributed across three categories: fire (14,300 images), smoke (14,300 images), and non-fire (14,300 images). The dataset is intentionally balanced to mitigate potential class bias during model training. In contrast, the WD dataset contains 2,700 images, categorized into five subclasses: smoke from fires, both fire and smoke, forested areas without confounding elements, fire confounding elements, and smoke confounding elements. This structure allows for a more nuanced analysis of fire and non-fire scenarios, particularly in distinguishing fire-resembling elements (e.g., sun glare, low-altitude clouds, or autumn foliage) from actual fire events.

However, certain dataset limitations must be acknowledged. FFS, despite its balanced class distribution, primarily sources images from online repositories, which may introduce regional biases and limit the dataset's

Fig. 6. Example of the dataset samples.

generalizability to global fire scenarios. WD, while designed for enhanced fire differentiation, exhibits an inherent class imbalance, with more non-fire images than fire-related ones. This could lead to a model bias toward non-fire classifications. Additionally, both datasets lack explicit metadata regarding environmental factors such as temperature, humidity, and wind speed, which could further refine fire detection performance. To mitigate these potential limitations, future work could incorporate additional real-world datasets, apply domain adaptation techniques, and integrate multimodal data sources (e.g., thermal imagery, meteorological data) to improve model robustness. The dataset is divided into a training set containing 31,696 images, a validation set with 14,259 images, and a test set comprising 7,896 images. This division not only provides ample samples for model training but also ensures robust performance evaluation at different stages, thereby enhancing the model's generalizability and practical application value. This partitioning strategy not only provides ample samples for training but also ensures comprehensive model evaluation across different development stages, enhancing its generalizability and practical application in wildfire detection systems.

To improve the model's generalization ability in complex fire scenarios, this study employed a series of data augmentation techniques. By applying various transformations to the images, the diversity of the training data was increased, reducing the risk of overfitting and enhancing the model's robustness. The specific data augmentation methods included rotation (random angles between 0° and 360°), flipping (horizontal and vertical), scaling, brightness adjustment, color transformation (contrast and saturation adjustments), and noise addition (Gaussian noise). These techniques simulate different shooting angles, directions, scales, lighting conditions, color variations, and image noise, thereby enhancing the model's adaptability to diverse environments and low-quality images. Furthermore, to label the fire and smoke targets in the images accurately, the LabelMe tool[61] was used for manual annotation, and the annotation results were converted to the YOLO format. The YOLO format is a streamlined object detection annotation format where each image's annotation information is stored in a corresponding .txt file. Each line in the file represents an object in the following format: `< class id >< x center >< y center >< width >< height> `. For example, "0.5265625 0.4171875 0.4921875 0.1765625" (Fig. 7a) and "1 0.4984375 0.165625 0.9875 0.3296875" (Fig. 7b)[61].

Model training

This study employs YOLOv11, the latest generation of object detection algorithms, which achieves fast inference speeds and efficient computational performance while maintaining high accuracy. The training was conducted on a Windows operating system with hardware equipped with three NVIDIA GeForce RTX 2080 Ti GPUs, each featuring 22GB of memory, 64GB of RAM, and an Intel i7 CPU. The system uses CUDA version 12.1, and the deep learning framework is PyTorch 2.1.2. The hardware configuration meets the demands of high-resolution images and large batch training, while multi-GPU parallel acceleration significantly reduces training time.

The training parameters are configured as follows: the total number of training epochs is set to 1000, supplemented by an early stopping mechanism. If the loss value does not significantly decrease over 10 consecutive epochs, training is terminated early to prevent overfitting. The batch size is set to 32, balancing GPU computational resources and training efficiency. This smaller batch size increases the update frequency, enabling the model to learn key features of flames and smoke more rapidly while avoiding memory overflow issues associated with larger batch sizes. The input image size is configured to 640×640 pixels, which reduces computational overhead without sacrificing image detail, allowing YOLOv11 to efficiently detect flames and smoke in complex scenes and maintain high detection speed and accuracy in practical applications.

Pretrained weights, specifically "YOLOv11x.pt," are utilized. These weights are pretrained on large-scale datasets, providing a robust initialization that significantly enhances detection and segmentation performance. The initial learning rate is set to 0.001, and a cosine annealing learning rate schedule is adopted to ensure that

the learning rate gradually decreases during the later stages of training. This approach facilitates smoother convergence and reduces potential oscillations. The Adam optimizer[62] is selected with parameters β_1=0.9, β_2=0.999, and ε = 1e-8, which accelerate the convergence process by adaptively adjusting the learning rate. Weight initialization is performed via He normal initialization[63] for the convolutional layers of the YOLOv11 model, ensuring rapid convergence and training stability. Additionally, L2 regularization (with a regularization coefficient of 0.0005) is applied to prevent overfitting and enhance the model's generalization ability.

Results evaluation and analysis

Evaluation and analysis of loss functions

In object detection tasks, model performance is typically assessed through the comprehensive optimization of multiple loss functions to ensure effective learning of various target attributes. This study conducted a thorough evaluation and analysis of different metrics during the training process on the basis of YOLOv11's multitask loss function. The loss function of YOLOv11 is composed of three components: bounding box loss (box loss), classification loss (cls loss), and distribution focal loss (dfl loss). Each component optimizes a distinct aspect of the model.

Box loss

Bounding box loss (box loss) quantifies the geometric disparity between the predicted bounding box and the ground truth bounding box, serving as a critical metric for optimizing the accuracy of object localization. Specifically, box loss assesses the differences in the center coordinates (x, y) and the width (w) and height (h) of the bounding boxes. By minimizing these discrepancies, the model enhances its prediction accuracy regarding the target object's central position and dimensions (Eq. (1))[64].

As depicted in Fig. 8, the box loss curves for the training and validation sets can be segmented into three distinct phases. First, during the early training phase (0–13 epochs), the initial box loss values for the training and validation sets are relatively high, at 1.5982 and 1.7223, respectively. This finding indicates that the model has a limited data fitting ability, resulting in inaccurate predictions and significant fluctuations. Second, in the middle training phase (epochs 14–351), as the model undergoes progressive optimization, the box loss for both the training and validation sets demonstrates a marked downward trend. The box loss for the training set gradually stabilizes, whereas the box loss of the validation set continues to decrease despite considerable fluctuations. This reflects an improvement in the model's generalization ability when handling diverse data. Finally, in the late training phase (epochs 352–501), box loss continues to decrease at a slower rate, with the training and validation sets' box loss values decreasing to 0.4827 and 0.8450, respectively. These reductions correspond to decreases of 69.79% and 50.94% from their initial values, respectively, indicating that the model achieves high accuracy and stability[64].

$$box\ loss = \sum_{i=1}^{n} \left(\left| x_i^{pred} - x_i^{gt} \right| + \left| y_i^{pred} - y_i^{gt} \right| + \left| w_i^{pred} - w_i^{gt} \right| + \left| h_i^{pred} - h_i^{gt} \right| \right) \quad (1)$$

——*where*:

x_i^{pred}, y_i^{pred} Center coordinates of the predicted bounding box;

x_i^{gt}, y_i^{gt} Center coordinates of the ground truth bounding box;

w_i^{pred}, h_i^{pred} Width and height of the predicted bounding box;

w_i^{gt}, h_i^{gt} Width and height of the ground truth bounding box;

a b

Fig. 7. Example of the labels of the dataset.

n Number of targets;

Cls loss

In object detection tasks, classification loss (cls loss) is a critical metric for evaluating the accuracy of the model's predicted target categories. This loss function assesses the discrepancies between the predicted class labels and true class labels, thereby optimizing the model's performance in classification tasks. In the YOLOv11 series models, (cls loss) is employed primarily to ensure that the model can accurately distinguish between target categories, such as by differentiating between flames and smoke in forest fire detection scenarios, as illustrated in Eq. 2.

$$cls\ loss = -\sum_{i=1}^{n} [y_i log(p_i) + (1 - y_i) log(1 - p_i)] \tag{2}$$

——*where*:

y_i True class label (value of 0 or 1, indicating whether the target belongs to a specific category);

p_i Probability predicted by the model that the target belongs to the category (ranging from 0 to 1);

n Number of categories;

These results demonstrate that YOLOv11 attains high classification accuracy and robust generalization capabilities in object detection tasks, effectively enhancing the model's applicability across various fire detection scenarios.

As illustrated in Fig. 9, the cls loss curves for both the training and validation sets can be analyzed in three distinct phases. Initially, during the early training phase (Epochs 0–25), the cls loss values for the training and validation sets start at 2.8058 and 1.7946, respectively. The cls loss of the training set peaks at 2.8058 during the first epoch, whereas the cls loss of the validation set reaches a higher peak of 4.6255 in the fourth epoch. This suggests that the model has not yet effectively learned discriminative features for classification, resulting in unstable performance. The greater fluctuation in the cls loss of the validation set further indicates a limited generalizability to unseen data at this stage.

In the middle training phase (Epochs 26–201), both training and validation cls loss values steadily decline, demonstrating that the model begins to effectively capture class-distinguishing features and improve classification performance. The continuous decrease in the training set's cls loss reflects the model's adaptation and fitting to the class patterns within the training data. Although the cls loss of the validation set exhibits some fluctuations in complex scenarios, the overall trend remains downward, signifying enhanced generalization performance.

Finally, in the late training phase (Epochs 201–501), the cls loss for both sets further decreases and stabilizes, with the training set's cls loss decreasing to 0.3242 and the validation set's cls loss to 0.5883. These values represent decreases of 88.44% and 67.22% from their initial values, respectively. This phase indicates that the model has achieved a well-converged state. Although the cls loss of the validation set is slightly greater than that of the training set, the minimal gap between them suggests that the model possesses strong generalizability when handling unseen data.

Dfl loss

Distribution focal loss (dfl loss) is introduced in object detection tasks to increase the prediction accuracy of bounding boxes. Unlike traditional discrete coordinate predictions, dfl loss models the bounding box

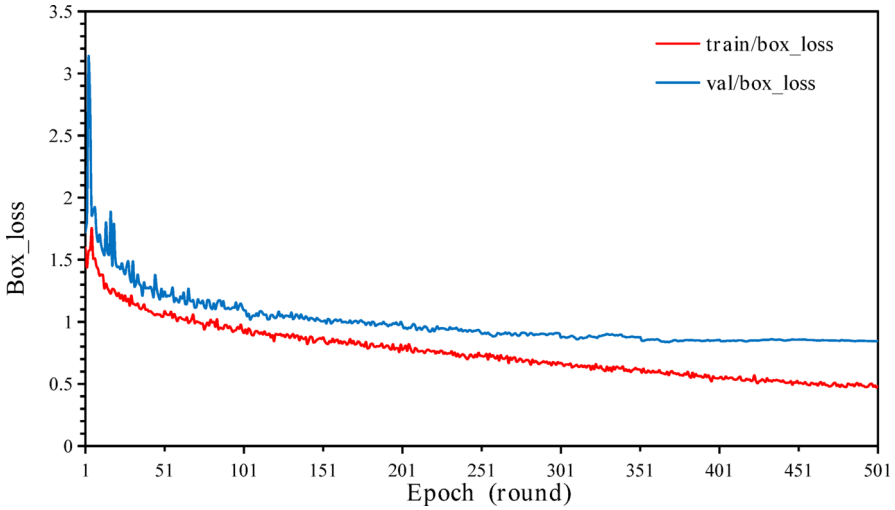

Fig. 8. Graph of the box loss curves on the training and validation sets.

coordinates as probability distributions and calculates the loss on the basis of these distributions. This approach optimizes the bounding box position at the subpixel level, significantly improving the prediction precision (Eq. 3) [84]. Specifically, dfl loss predicts a continuous coordinate distribution for each bounding box boundary (such as the left, right, top, and bottom boundaries) and optimizes these distributions. This enables the model to more accurately adjust irregular and blurry boundaries of flames and smoke, thereby enhancing detection performance. The formula is as follows:

$$dfl\ loss = -\sum_{i=1}^{n}\sum_{j=1}^{k}[w_j.\ y_{ij}.\ log(p_{ij})] \tag{3}$$

——*where*:

$\boldsymbol{n}$ Number of bounding box boundaries (e.g., left, right, top, bottom) in an image, with four bounding boxes per box.

$\boldsymbol{k}$ The number of distribution intervals obtained after discretizing each boundary position;

$\boldsymbol{y_{ij}}$ Discrete value at the *j*-th interval of the *i*-th boundary after discretizing the actual bounding box position;

$\boldsymbol{p_{ij}}$ The probability predicted by the model for the *j*-th value at the *i*-th boundary position;

$\boldsymbol{w_j}$ The weight value used to adjust the distribution accuracy.

As illustrated in Fig. 10, the dfl loss curves for the training and validation sets align with the trends observed in box loss and cls loss and can be divided into three stages:

Early training phase (epochs 0–45): The initial dfl loss values are 2.4556 (training set) and 2.7088 (validation set). The training set's cls loss peaks at 3.1271 in the 5th epoch, whereas the validation set's cls loss reaches 4.1895 in the 3rd epoch, indicating instability in class discrimination and weaker generalizability to unseen data.

Middle training phase (epochs 46–351): As training progresses, dfl loss decreases significantly for both datasets, reflecting improvements in learning distribution characteristics and optimizing classification and detection. The training set's dfl loss steadily declines, whereas the validation set shows fluctuations in complex scenarios but maintains an overall downward trend, indicating enhanced generalizability. During the late training phase (epochs 351–501), dfl loss further decreases and stabilizes, reaching 1.1241 (training set) and 1.6099 (validation set), corresponding to reductions of 54.22% and 40.57% from the initial values and 64.05% and 61.57% from the peak values, respectively. The model achieves a well-converged state, with a small gap between training and validation losses, indicating strong generalizability to unseen data. This trend highlights dfl loss's role in achieving high-precision detection, particularly in complex scenarios such as flames and smoke, enhancing YOLOv11's applicability in diverse object detection tasks.

To further support the visual interpretation of the training curves, Table 3 provides a comprehensive numerical summary of the three core loss functions—box loss, classification loss, and distribution focal loss—on both the training and validation sets. The table presents the initial values, maximum values (with corresponding training rounds), final values, and relative drop rates. These quantitative metrics reinforce the graphical findings, demonstrating a consistent and substantial decrease across all loss components as training progresses. Notably, the classification loss on the validation set decreased by over 67%, and the box loss showed a reduction of nearly 70% on the training set, indicating effective convergence and improved generalization capability.

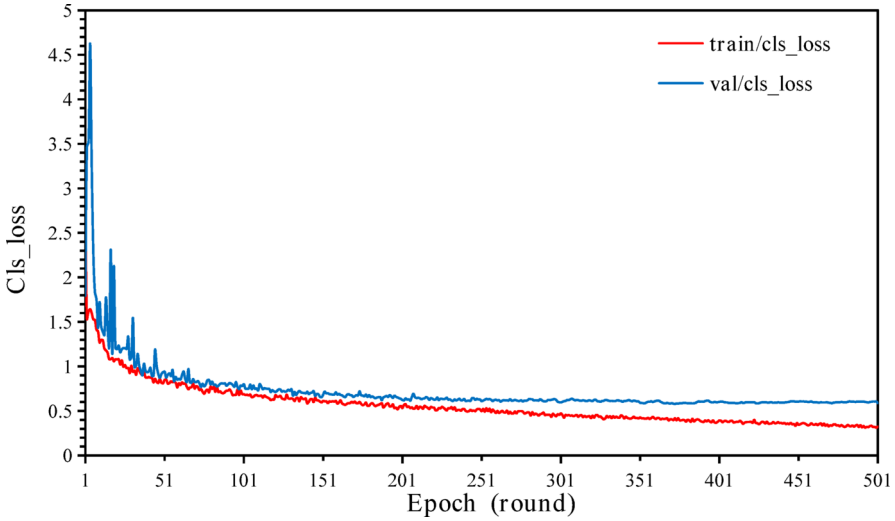

Fig. 9. Graph of the cls loss curves on the training and validation sets.

	box_loss		cls_loss		dfl_loss	
	train	val	train	val	train	val
Initial Value	1.5982	1.7223	2.8058	1.7946	2.4556	2.7088
Maximum Value	1.7548 (round 6)	3.1304 (round 3)	2.8058 (round 1)	4.6255 (round 4)	3.12718 (round 6)	4.1895 (round 3)
Final Value	0.4827	0.8450	1.1241	0.5883	1.1241	1.6099
Drop Rate from Initial Value	69.79%	50.94%	88.44%	67.22%	54.22%	40.57%
Drop Rate from Maximum Value	72.49%	73.01%	88.44%	87.28%	64.05%	61.57%

Table 3. Summary of training and validation losses for box loss, classification loss, and distribution focal loss, including initial, maximum, and final values, as well as percentage reductions.

Evaluation and analysis of the accuracy metrics

In object detection tasks, commonly used accuracy metrics for evaluating model performance include precision, recall, mAP < sub > 50</sub> (mean average precision at an IoU threshold of 50%), and mAP < sub > 50–95</sub> (mean average precision across multiple IoU thresholds). These metrics assess the model's accuracy in locating and classifying targets, coverage, and overall detection capability. Precision is defined as the proportion of true positive predictions among all positive predictions made by the model (Eq. 4). Recall measures the proportion of true positive detections out of all actual positive samples (Eq. 5). mAP < sub > 50</sub > represents the model's average precision at an IoU threshold of 0.50, calculated as (Eq. 6)[65]. mAP < sub > 50–95</sub > is the average precision calculated across multiple IoU thresholds ranging from 0.50 to 0.95 with a step size of 0.05, as shown in Eq. (7)[65].

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$mAP_{50} = \frac{1}{N} \sum {}^{N}_{i=1} AP_{50,i} \tag{6}$$

$$mAP_{50-95} = \frac{1}{N} \sum {}^{N}_{i=1} \frac{1}{K} \sum {}^{K}_{j=1} AP_{i,j} \tag{7}$$

——*where*:

TP (true positive): the number of samples correctly predicted as positive.

FP (false positive): the number of samples incorrectly predicted as positive.

FN (false negative): the number of samples incorrectly predicted as negative.

N: Total number of target categories.

$AP_{50,i}$ Average precision for category i at an IoU threshold of 0.5.

K Number of different IoU thresholds (10 values from 0.5 to 0.95).

$AP_{i,J}$ Average precision for category *i* at the *j*-th IoU threshold.

Figure 11 shows the curves of different metrics on the training and validation sets. The precision initially decreases from 0.664 to 0.076 during the early training phase (epochs 0–25) and then gradually increases to 0.949, indicating a reduction in the model's false positive rate and a significant improvement in classification accuracy. Recall decreases from an initial value of 0.619 to 0.193 and then steadily increases to 0.850, reflecting a substantial enhancement in the model's ability to capture positive samples and a significant reduction in false negatives. mAP < sub > 50</sub > decreases from 0.668 to 0.048 in the early training phase and then increases to 0.901, demonstrating a notable improvement in the model's detection accuracy at a single IoU threshold. Similarly, mAP < sub > 50–95</sub > decreases from 0.408 to 0.017, followed by an increase to 0.786, indicating a considerable improvement in the model's comprehensive detection capability across multiple thresholds, especially under high IoU thresholds.

Overall, all four metrics initially decrease but then increase, reflecting the model's continuous optimization of feature extraction and classification capabilities during training. This progression gradually enhances the overall detection performance and generalizability of the model. Precision and recall directly indicate the model's fundamental classification capabilities, whereas mAP < sub > 50</sub > and mAP < sub > 50–95</sub > provide a more comprehensive evaluation of the model's detection performance under varying degrees of overlap. In particular, mAP < sub > 50–95</sub > imposes greater demands on the model's comprehensive detection ability, underscoring its effectiveness in diverse scenarios.

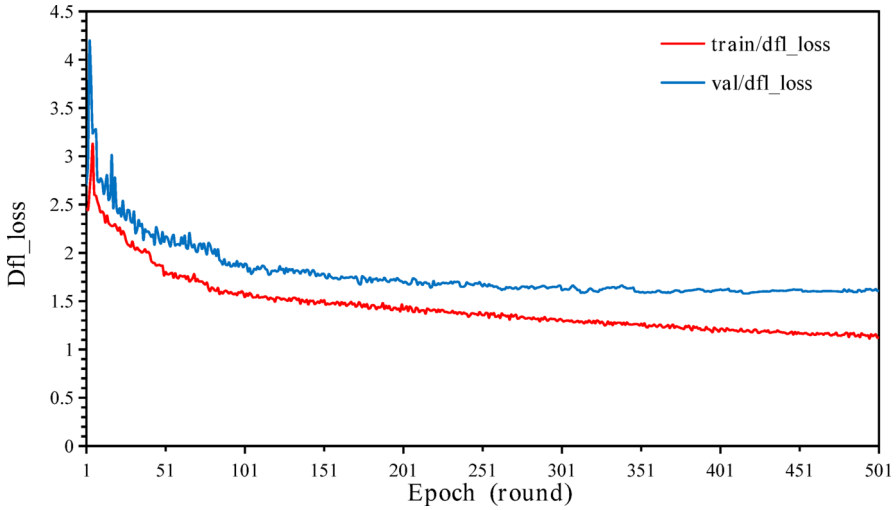

Fig. 10. Graph of the dfl loss curves on the training and validation sets.

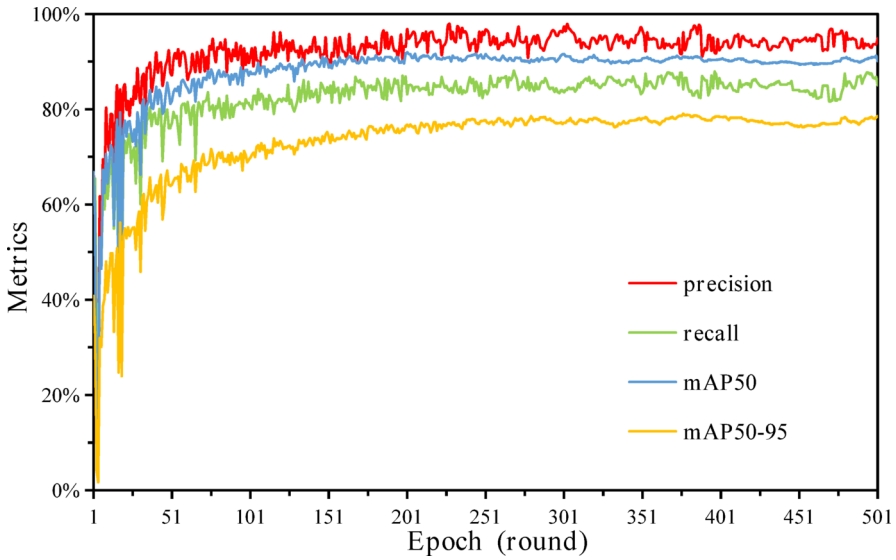

Fig. 11. The various accuracy curves of the metrics.

Evaluation and analysis of F1 score

In the research and application of deep learning-based forest fire smoke recognition object detection technology, F1 Score is a key metric for evaluating classifier performance, particularly with imbalanced datasets (e.g., in forest fire detection, smoke occurrences are much less frequent than background images). The F1 Score is the harmonic mean of precision and recall, providing a balanced evaluation of model performance by considering both false positives and false negatives. Its formula is shown in Eq. 8.

$$F1\ Score = 2 \times \frac{Precision \times\ Recall}{Precision + Recall} \tag{8}$$

Analyzing the trend of the curve, the model's performance fluctuated significantly in the early stages (the first 10 epochs), with a sharp drop to a minimum value of 0.1089 in the 4th epoch (Figure 12). This could be due to unstable initial model weights or performance degradation caused by an excessively high learning rate. However, after this stage, the model quickly recovered and improved, demonstrating its ability to capture key data features in a short time. After 50 epochs, the F1 Score stabilized above 0.8 and gradually became steady. By approximately 200 epochs, the F1 Score showed minimal fluctuation, indicating that the model was nearing convergence. By the end of training, the final F1 Score reached 0.8968, marking a 39.96% improvement from the initial value and a dramatic 723.72% increase from the 4th epoch's minimum value. This highlights the model's exceptional smoke detection ability after training. In the context of forest fire smoke recognition, an F1 Score close to 0.9 indicates

	precision	recall	mAP50	mAP50-95	F1 Score
Initial Value	0.6637	0.6192	0.6679	0.4081	0.6407
Maximum Value	0.0758 (round 4)	0.1928 (round 4)	0.04802 (round 4)	0.0166 (round 4)	0.1089 (round 4)
Final Value	0.9487	0.8501	0.9010	0.7856	0.8968
Rise Rate from Initial Value	42.94%	37.28%	34.88%	92.49%	39.96%
Rise Rate from Maximum Value	1151.07%	340.76%	1776.09%	4631.09%	723.72%

Table 4. Summary of key performance metrics (precision, recall, mAP@0.5, mAP@0.5:0.95, and F1 Score) during training, including initial, minimum, and final values, and percentage improvements.

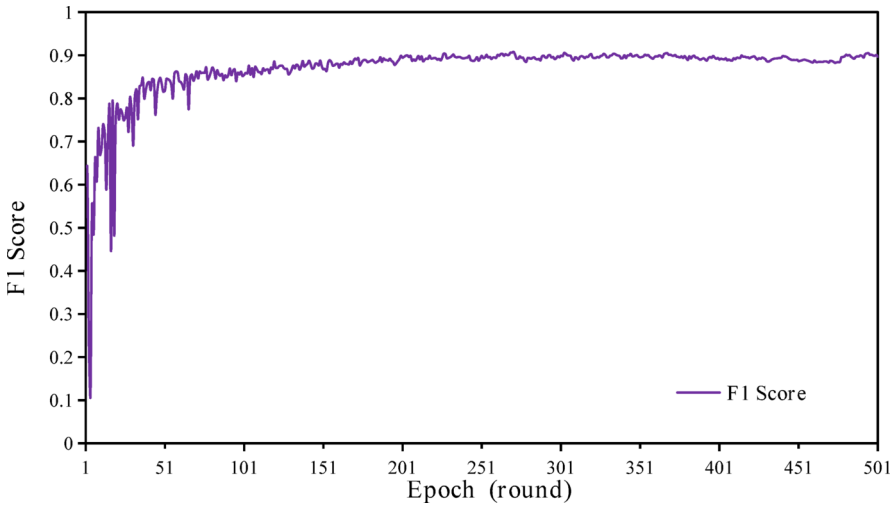

Fig. 12. F1 Score plot.

high detection precision and recall, effectively minimizing the risks of false positives and false negatives, and meeting the dual requirements of high accuracy and real-time performance for practical applications. Early performance fluctuations may be due to factors such as uneven data distribution, initial weight settings, or a high learning rate. As training progressed, the model adapted to the complex features of forest environments, exhibiting strong robustness in smoke detection.

As a complement to the F1 score and accuracy trend analyses presented earlier, Table 4 summarizes the numerical progression of major evaluation metrics, including precision, recall, mAP@0.5, mAP@0.5:0.95, and F1 Score. For each metric, the table lists the initial values, observed minimums during early epochs (where applicable), and the final values upon training completion. Additionally, percentage improvements from both the initial and lowest values are calculated. The F1 Score improved by nearly 40% from its initial state and over 720% from its minimum, while mAP@0.5:0.95 nearly doubled, reflecting a significant boost in detection robustness and consistency. These results further validate the effectiveness of the YOLOv11 model in handling complex detection tasks such as forest fire smoke recognition.

Evaluation and analysis of precision-recall curves

In this study, the precision-recall curve (PR curve) is utilized to illustrate the trade-off between precision and recall under varying threshold conditions during the detection of flames or smoke. By adjusting the detection threshold, the values of precision and recall change, enabling the plotting of a comprehensive PR curve. Typically, the shape of the PR curve provides an intuitive assessment of the model's performance: a curve positioned in the upper right corner and resembling a square indicates excellent performance in both precision and recall, whereas a curve situated in the lower left suggests poor model performance. By analyzing the shape of the PR curve, the model's performance across different thresholds can be evaluated, and the optimal threshold can be selected to optimize the prediction results. Furthermore, the area under the PR curve represents the average precision (AP) for that category. mAP@0.5 measures the overall performance of the model by averaging the AP values across multiple categories at an intersection over union (IoU) threshold of 0.5. Specifically, under the condition of $IoU \geq 0.5$, mAP@0.5 assesses the model's comprehensive performance across different categories by calculating the mean of the areas under each category's PR curve. A higher mAP@0.5 value signifies stronger overall performance in detection tasks. As depicted in Fig. 13, the evaluation of the PR curve reveals that the model achieved an average mAP@0.5 of 0.901, demonstrating high detection accuracy. Although the mAP@0.5 values vary among different categories, the smoke category attained an mAP@0.5 of 0.962, whereas the flame category reached 0.841. Notably, despite the slightly lower precision of the flame category than the smoke

category, an mAP@0.5 of 0.841 represents outstanding performance in flame detection. Flames present high detection difficulty because of their variable shapes, complex features, and strong dynamics. Nonetheless, the model maintains high precision in detecting such complex targets, underscoring its robustness and recognition capabilities. These results indicate that YOLOv11 exhibits excellent detection performance and has significant application potential across various fire scenarios.

Comprehensive evaluation and analysis of test set results

Confidence is a key metric for assessing detection reliability, indicating the model's confidence in the presence of a target and its classification within the predicted bounding box. The definition of confidence, shown in Eq. 11, is determined by two primary factors: the predicted probability of the target class and the alignment between the bounding box and the target's true position. It represents the model's confidence in predicting a specific target, derived from the training data as a probability value, typically ranging from 0 to 1. The closer the confidence value is to 1, the more confident the model is in detecting the target; conversely, the closer the confidence value is to 0, the more the model believes there is no target in the image or that the prediction is unreliable.

$$Confidecne = P\left(Object\right) \times \ Clsaa\ Probability \tag{11}$$

——where:

P(Object) The probability that an object exists within the predicted bounding box, indicating the model's confidence that the detection box contains an object.

Class probability The likelihood that the detected object belongs to a specific category, assuming the object is present.

In object segmentation tasks, confidence typically represents the model's degree of certainty regarding a particular detection result, quantified as the probability that the model considers the detection to be correct. Higher confidence implies greater certainty in the accuracy of the detection results, which is usually associated with higher precision, indicating a lower false positive rate at high confidence levels. Conversely, lower confidence suggests less certainty in the model's judgments, increasing the likelihood of false positives or false negatives. Therefore, confidence serves as a crucial metric not only for evaluating the model's certainty in individual detection results but also for reflecting the model's overall detection performance across different confidence thresholds.

As illustrated in Fig. 14, within the test set, 86.89% of the samples have confidence scores exceeding 0.85, indicating that the model is highly confident in the detection results of the majority of samples. However, performance varies across different target types (flames and smoke). The average confidence for flame detection is 0.90, whereas for smoke detection, it is 0.88, indicating that the model is more certain in detecting flames than in detecting smoke. To determine if these differences are statistically significant, a t-test was conducted on the confidence scores for flames and smoke. The results show that the difference in average confidence between flames (72.21%) and smoke (78.87%) is statistically significant ($p < 0.05$), indicating that the model is significantly more confident in detecting smoke than flames. This difference may stem from the distinct visual characteristics of flames and smoke: flames typically possess more defined shapes and color features, whereas smoke exhibits more uncertain and variable shapes and textures. This variability poses greater challenges for the model in accurately detecting smoke, resulting in slightly lower confidence scores. These findings suggest that although the model demonstrates high overall confidence in detections, there are subtle differences in performance across different target types. This highlights the necessity for further optimization of the model to increase the detection accuracy for complex targets such as smoke.

Application case testing

To test the object detection model, this study utilized the publicly available "Forest Fire Classifier Dataset" [86] to comprehensively evaluate and validate the model's performance. This dataset comprises high-definition images that encompass a wide range of forest fire scenarios, including various terrains (such as mountainous regions, plains, dense forests, etc.) and diverse meteorological conditions (such as sunny days, cloudy days, strong winds, etc.). It effectively simulates the myriad situations that may be encountered during fire detection, providing a diverse set of test samples to ensure the model's robustness in complex environments. The intricate nature of forest fire scenes presents significant challenges to fire detection models, necessitating not only the accurate identification of flame and smoke features but also the maintenance of high detection accuracy under varying lighting and occlusion conditions. Furthermore, the rapid spread of fires and the dispersion of smoke increase the complexity of object segmentation tasks. Consequently, the "Forest Fire Classifier Dataset" offers a highly challenging testing platform for assessing the model's generalizability and robustness. During the application testing phase, the model demonstrated commendable performance in the object segmentation tasks for both flames and smoke.

As depicted in Fig. 15, the test results indicate overall strong object detection performance, with particularly robust outcomes in the flame detection task. This discrepancy in performance can be attributed to the distinct visual features of flames and smoke: flames exhibit more pronounced colors and shapes, making them easier for the model to recognize, whereas smoke presents more ambiguous and variable textures and shapes, imposing greater demands on the model's detection capabilities. The experimental results reveal that the model can deliver relatively accurate detection outcomes in the majority of cases, providing reliable support for fire warning systems. These findings further substantiate that the object detection model not only excels in controlled

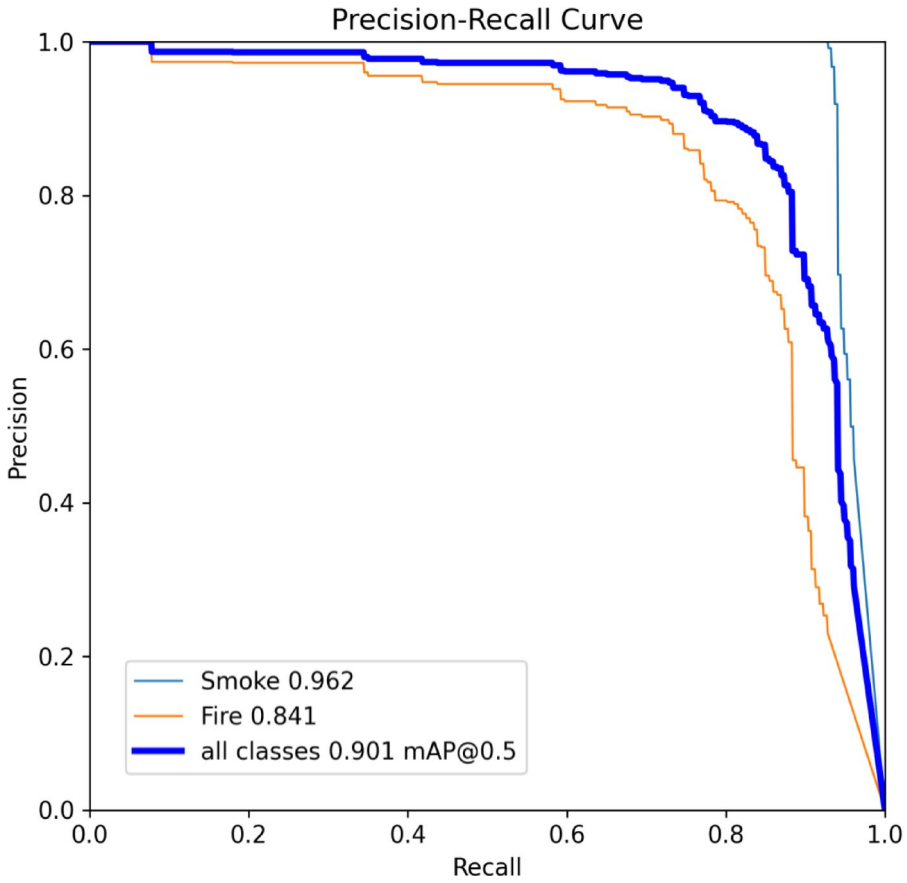

Fig. 13. Precision–recall curves.

Fig. 14. Example of the test set samples.

Fig. 15. Application case test.

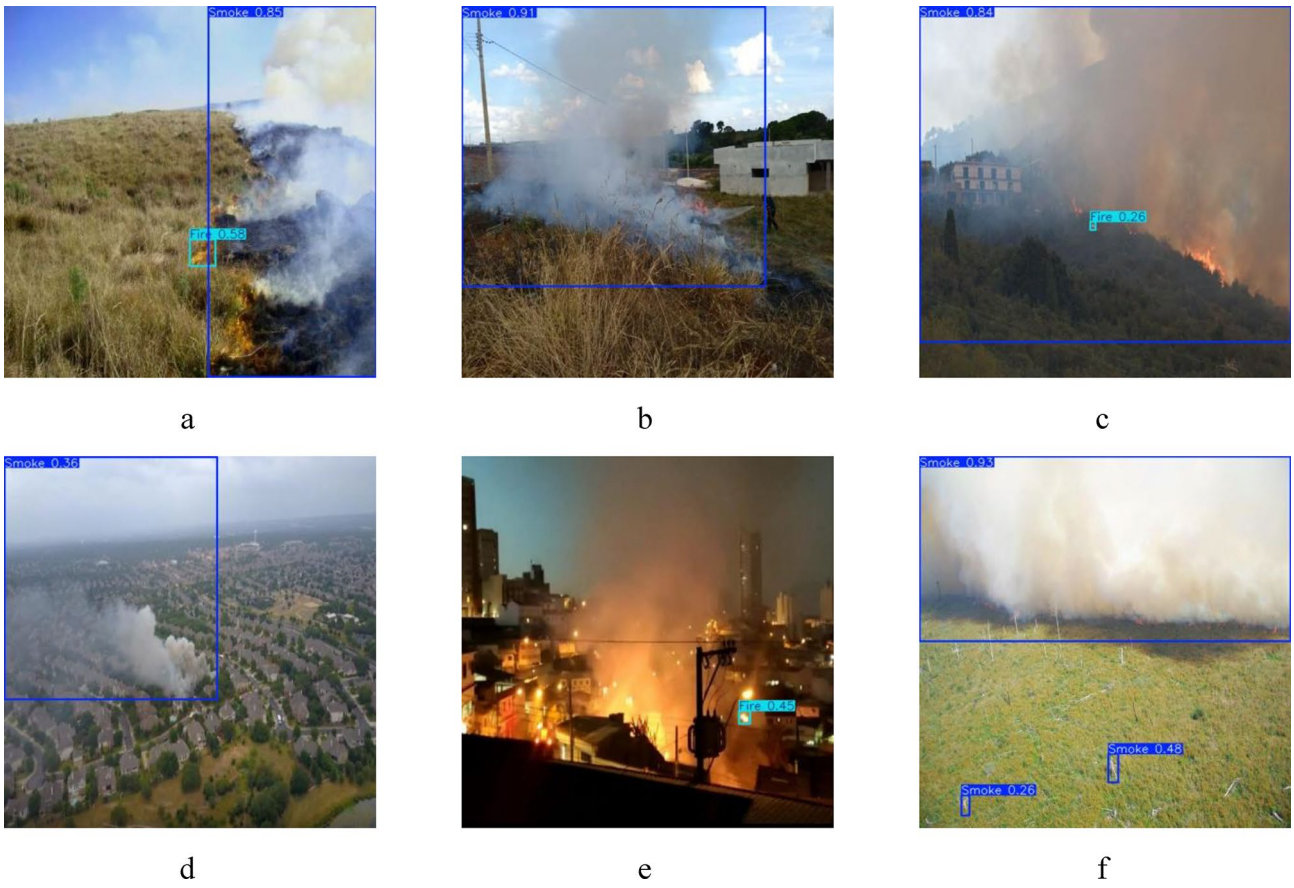

Fig. 16. Images with identified issues.

experimental environments but also demonstrates excellent generalizability for detecting flames and smoke in real-world fire scenarios.

Discussion

Error case analysis

This study analyzes and evaluates the application of the YOLOv11 model in different fire scenarios. The results indicate that the model demonstrates promising performance in real-world fire monitoring. However, YOLOv11 faces significant challenges in certain complex scenarios.

First, the model has limitations when dealing with cases where the visual features of flames and smoke overlap. In Fig. 16a and b, although the visual characteristics of both flames and smoke are clearly visible, the model fails to correctly identify some flame areas, resulting in lower detection confidence. This missed detection is mainly due to the small and blurry nature of the flame, along with its overlap with smoke, preventing the model from accurately extracting flame features, particularly in the early stages of a fire. Figure 16c shows a failure in fire detection, where, despite the model successfully detecting smoke (confidence 0.84), the detection confidence for the flame is low (0.26), and the model fails to identify the obvious flame region. This highlights the issue of false negatives (missed detections), where the model fails to detect the flame despite its clear presence. The flame's color and shape may overlap significantly with the background or surrounding smoke, making it difficult for the model to capture the flame's details, especially when the flame is distant or when heavy smoke causes the flame's edges to become blurry, complicating the model's ability to distinguish the flame area (Fig. 16e).

Moreover, the impact of low contrast and complex backgrounds poses another significant challenge. Figure 16d illustrates a failure in smoke detection under low contrast. Although the model detects smoke, the low image contrast reduces the confidence to 0.36, indicating that the model struggles to extract smoke features accurately in such conditions. In complex urban fire scenarios, background elements like buildings, streetlights, and power lines, which resemble the color and shape of smoke, complicate the recognition task, causing the model to be more susceptible to interference and leading to misjudgments or missed detections.

Finally, flames, being dynamic small targets, are particularly challenging to detect, especially in long-distance shots or in the early stages of a fire, where their shape and edges are often blurry. Figure 16f illustrates a missed detection of a small flame. At the same time, the false positive issue in Fig. 16f reveals the difficulty the model faces in distinguishing between background objects and fire targets. The model mistakenly identifies a branch as smoke, indicating the model's insufficient ability to separate background elements from fire targets. At long distances or low resolutions, branches' color and shape closely resemble smoke features, and the boundaries between background objects and fire targets become blurred, leading to misidentifications. Natural elements such as branches and grass, which resemble smoke in color or shape, increase the likelihood of incorrect identification.

In conclusion, although the YOLOv11 model shows promising performance in fire monitoring applications, it faces several challenges in complex scenarios. The overlap of visual features between flames and smoke, difficulties in low-contrast environments, and interference from complex backgrounds are major limitations of the current model. Particularly in scenarios with small targets, dynamic targets, and complex backgrounds, the model exhibits issues with both false negatives (missed detections) and false positives (misdetections). These challenges underscore the model's insufficient adaptability in handling diverse and complex fire environments.

Future directions for optimization and improvement

This study highlights the potential of the YOLOv11 model for fire monitoring, but it still faces limitations in complex scenarios such as the overlap of flame and smoke visual features, low-contrast environments, interference from complex backgrounds, and small target detection. To address these challenges, future research should focus on several optimization areas to enhance the model's robustness and accuracy.

Firstly, multi-modal data fusion will be a crucial direction for improving YOLOv11's performance. By integrating thermal imaging with other data sources, the model can effectively distinguish flames from smoke based on temperature information, even in dense smoke or when flame boundaries are blurred. Thermal imaging compensates for the shortcomings of visual images, enhancing flame detection in low-contrast or complex background settings. For example, thermal imaging enables fire detection in low-light or smoky conditions, reducing the interference of visual data[66]. Additionally, incorporating self-attention mechanisms, such as Transformers or attention modules, can enhance the model's focus on key areas, improving its accuracy in complex environments, particularly when flames and smoke overlap heavily[67,68].

For low-contrast and complex background environments, future research should prioritize improving image enhancement techniques, especially contrast and brightness adjustments, to better extract features from low-contrast images[69,70]. Background separation is also a critical optimization avenue. Strengthening background modeling will help reduce the impact of complex environments—such as urban landscapes or forest backdrops—on fire detection, improving model accuracy[71]. For instance, in Fig. 16e, the urban background's similarity to smoke caused the model to misidentify the scene. Background separation will aid in more accurate fire detection by reducing misdetections and missed detections.

Furthermore, multi-scale detection will enhance the model's ability to detect small flame targets, especially in early fire stages or at long distances[72]. While YOLOv11 has made improvements in small target detection, issues with missing small flames persist. Multi-scale feature extraction will enable the model to process targets at various scales, improving small target detection accuracy[72]. By integrating feature maps from different scales, the model can more effectively identify flames of varying sizes, thus boosting its overall detection capabilities.

Finally, interdisciplinary collaboration and the integration of intelligent monitoring systems will shape future research. By incorporating insights from fields like ecology and climatology, deep learning models can better predict fire spread and improve early warning systems[73,74]. Combining YOLOv11 with technologies like drones, satellite remote sensing, and ground sensors will facilitate the development of comprehensive fire monitoring systems, enabling real-time global monitoring and rapid, accurate responses[75–78]. Through sensor data fusion and multi-task learning, the model will extract information from diverse data sources, significantly improving the efficiency and accuracy of fire monitoring.

In summary, YOLOv11 and its future optimized versions hold tremendous potential in fire monitoring. By incorporating multi-modal data fusion, enhancing image processing techniques, expanding dataset diversity, and improving real-time monitoring capabilities, the model will deliver more accurate and robust performance in complex environments. In particular, future work will involve the integration of real-time datasets such

as FLAME and UAVS-FFDB, which will help validate the model under realistic and time-sensitive wildfire scenarios. These efforts will provide strong technical support for global fire prevention, early warning systems, and emergency responses.

Conclusion

This study investigates the application of the YOLOv11 algorithm for forest fire smoke recognition, leveraging deep learning-based object detection techniques. Experiments were conducted using two publicly available fire image datasets—WD and Forest Fire Smoke (FFS)—and after 501 training epochs, the model was comprehensively evaluated using various metrics. The results based on loss functions indicate that the model performs well across multiple dimensions, including bounding box loss, classification loss, and distribution focal loss, demonstrating effective optimization of object detection performance. Further evaluation using accuracy metrics validated the model's robustness, achieving a precision of 0.949, a recall of 0.850, an mAP@0.5 of 0.901, and an mAP@0.5:0.95 of 0.786. In particular, the mAP@0.5 for the smoke category reached 0.962, highlighting the model's superior ability to detect smoke, although the flame category, with an mAP@0.5 of 0.841, presented slightly lower performance due to its complex and dynamic visual characteristics. Additionally, 86.89% of the test samples achieved confidence scores above 0.85, reflecting high detection reliability and consistency. While the YOLOv11 algorithm demonstrates excellent performance in forest fire smoke detection, some limitations remain. In particular, the model occasionally struggles to distinguish between overlapping smoke and flame regions, and its detection accuracy may decrease in low-contrast or cluttered backgrounds. Furthermore, the current datasets, while diverse, may not fully represent the range of real-world scenarios encountered in active wildfire conditions.

Future work will focus on addressing these limitations by incorporating additional real-time and high-resolution datasets such as FLAME and UAVS-FFDB to further enhance model generalization and robustness. Moreover, the integration of multi-modal data sources—such as thermal infrared imagery, meteorological data, and topographic context—will be explored to improve the detection of obscured or small-scale fire events. Enhancements to the model's multi-scale detection capabilities and the adoption of advanced attention mechanisms will also be considered to improve the identification of visually complex targets. These efforts aim to support the development of a more adaptive, accurate, and deployable intelligent forest fire monitoring system.

Data availability

The data presented in this study are available upon request from the corresponding author.

Received: 16 January 2025; Accepted: 9 April 2025
Published online: 10 May 2025

References

1. Gajendiran, K., Kandasamy, S. & Narayanan, M. Influences of wildfire on the forest ecosystem and climate change: a comprehensive study. *Environ. Res.* **240**, 117537 (2024).
2. Arteaga, B., Diaz, M. & Jojoa, M. Deep learning applied to forest fire detection. 2020 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). IEEE. pp. 1–6. (2020).
3. Kalogiannidis, S., Chatzitheodoridis, F., Kalfas, D., Patitsa, C. & Papagrigoriou, A. Socio-psychological, economic and environmental effects of forest fires. *Fire* **6** (7), 280 (2023).
4. Shang, D. et al. Deep learning-based forest fire risk research on monitoring and early warning algorithms. *Fire* **7** (4), 151 (2024).
5. Guan, Z. et al. Forest fire segmentation from aerial imagery data using an improved instance segmentation model. *Remote Sens. (Basel)*. **14** (13), 3159 (2022).
6. Wang, G., Wang, F., Zhou, H. & Lin, H. Fire in focus: advancing wildfire image segmentation by focusing on fire edges. *Forests* **15** (1), 217 (2024).
7. Niu, K. et al. An improved YOLOv5s-seg detection and segmentation model for the accurate identification of forest fires based on UAV infrared image. *Remote Sens. (Basel)*. **15** (19), 4694 (2023).
8. Karim, S. & Hayawi, M. J. Fire detection by using CNN Alex net algorithm. *J. Educ. Pure Science-University Thi-Qar* **14**(1), 70–78 (2024).
9. Tan, M. & Bakır, H. Enhancing forest fire detection: utilizing VGG16 for high-accuracy image classification and machine learning integration. 2024 8th International Artificial Intelligence and Data Processing Symposium (IDAP). IEEE. pp. 1–9. (2024).
10. Biswas, A., Ghosh, S. K. & Ghosh, A. Early fire detection and alert system using modified inception-v3 under deep learning framework. *Procedia Comput. Sci.* **218**, 2243–2252 (2023).
11. Muhammad, S. S. & Alrikabi, J. M. Fire detection by using densenet 201 algorithm and surveillance cameras images. *J. Al-Qadisiyah Comput. Sci. Math.* **16** (1), 81–91 (2024).
12. Barkunan, S. R., Raja, S. R., Kannagi, L., Maniraj, P. & Venu, N. Mobile blaze: harnessing mobile net architecture for efficient forest fire detection. 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies. IEEE. pp. 1–7. (2024).
13. Akilandeswari, A., Amanullah, M., Nanthini, S., Sivabalan, R. & Thirumalaikumari, T. Comparative study of fire detection using SqueezeNet and VGG for enhanced performance. 2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM). IEEE. pp. 1–6. (2024).
14. Ilyas, B. R. et al. Forest fire detection with combined SVM and deep CNN approach. 2024 2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC). IEEE. pp. 1–6. (2024).
15. Reis, H. C. & Turk, V. Detection of forest fire using deep convolutional neural networks with transfer learning approach. *Appl. Soft Comput.* **143**, 110362 (2023).
16. Guede-Fernández, F., Martins, L., de Almeida, R. V., Gamboa, H. & Vieira, P. A deep learning based object identification system for forest fire detection. *Fire* **4** (4), 75 (2021).
17. Khan, S. & Khan, A. Ffirenet: deep learning based forest fire classification and detection in smart cities. *Symmetry* **14** (10), 2155 (2022).
18. Zhang, L., Wang, M., Fu, Y. & Ding, Y. A forest fire recognition method using UAV images based on transfer learning. *Forests* **13** (7), 975 (2022).

19. Hu, X., Ban, Y. & Nascetti, A. Uni-temporal multispectral imagery for burned area mapping with deep learning. *Remote Sens. (Basel)*. **13** (8), 1509 (2021).
20. Siddique, N., Paheding, S., Elkin, C. P. & Devabhaktuni, V. U-net and its variants for medical image segmentation: a review of theory and applications. *IEEE Access*. **9**, 82031–82057 (2021).
21. Ozturk, O., Sarıtürk, B. & Seker, D. Z. Comparison of fully convolutional networks (FCN) and u-net for road segmentation from high resolution imageries. *Int. J. Environ. Geoinformatics*. **7** (3), 272–279 (2020).
22. Azad, R., Asadi-Aghbolaghi, M., Fathy, M. & Escalera, S. Attention deeplabv3+: multi-level context attention mechanism for skin lesion segmentation. European conference on computer vision. Springer. pp. 251 – 66. (2020).
23. Zhang, C., Lu, W., Wu, J., Ni, C. & Wang, H. SegNet network architecture for deep learning image segmentation and its integrated applications and prospects. *Acad. J. Sci. Technol.* **9** (2), 224–229 (2024).
24. Chai, H., Yan, C., Zou, Y. & Chen, Z. Land cover classification of remote sensing image of Hubei Province by using PSP net. *Geomatics Inform. Sci. Wuhan Univ.* **46** (8), 1224–1232 (2021).
25. Shirvani, Z., Abdi, O. & Goodman, R. C. High-resolution semantic segmentation of woodland fires using residual attention UNet and time series of sentinel-2. *Remote Sens. (Basel)*. **15** (5), 1342 (2023).
26. Wang, G. et al. RFWNet: a multi-scale remote sensing forest wildfire detection network with digital twinning, adaptive spatial aggregation, and dynamic sparse features. *IEEE Trans. Geosci. Remote Sens.* **62**(5), 1234–1245 (2024).
27. Yang, S., Huang, Q. & Yu, M. Advancements in remote sensing for active fire detection: a review of datasets and methods. *Sci. Total Environ.* **943**, 173273 (2024).
28. Zhang, L. et al. FBC-ANet: a semantic segmentation model for UAV forest fire images combining boundary enhancement and context awareness. *Drones-Basel* **7** (7), 456 (2023).
29. Kaur, J. & Singh, W. Tools, techniques, datasets and application areas for object detection in an image: a review. *Multimed Tools Appl.* **81** (27), 38297–38351 (2022).
30. Cao, X., Su, Y., Geng, X. & Wang, Y. YOLO-SF: YOLO for fire segmentation detection. *IEEE Access.* **11**, 111079–111092 (2023).
31. Wang, T. et al. Improving YOLOX network for multi-scale fire detection. *Visual Comput.* **40** (9), 6493–6505 (2024).
32. Liu, H. et al. Tfnet: transformer-based multi-scale feature fusion forest fire image detection network. *Fire* **8** (2), 59 (2025).
33. Zhang, J. Enhancing tree type detection in forest fire risk assessment: multi-stage approach and color encoding with forest fire risk evaluation framework for UAV imagery. *Arxiv Preprint Arxiv*, 2407. https://doi.org/10.48550/arXiv.2407.19184 (2024).
34. Sun, P. et al. Sparse r-CNN: an end-to-end framework for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **45** (12), 15650–15664 (2023).
35. Li, Z., Li, E., Xu, T., Samat, A. & Liu, W. Feature alignment FPN for oriented object detection in remote sensing images. *IEEE Geosci. Remote Sens. Lett.* **20**, 1–5 (2023).
36. Li, R. et al. Automatic Bridge crack detection using unmanned aerial vehicle and faster r-CNN. *Constr. Build. Mater.* **362**, 129659 (2023).
37. Diwan, T., Anirudh, G. & Tembhurne, J. V. Object detection using YOLO: challenges, architectural successors, datasets and applications. *Multimed Tools Appl.* **82** (6), 9243–9275 (2023).
38. Wang, H. et al. Centernet-auto: a multi-object visual detection algorithm for autonomous driving scenes based on improved Centernet. *IEEE Trans. Emerg. Top. Comput. Intell.* **7** (3), 742–752 (2023).
39. Zhang, Y. et al. Multi-view feature-based {SSD} failure prediction: what, when, and why. 21st USENIX Conference on File and Storage Technologies (FAST 23). pp. 409 – 24. (2023).
40. Yusuf, M. O. et al. Target detection and classification via efficientdet and CNN over unmanned aerial vehicles. *Front. Neurorobot.* **18**, 1448538 (2024).
41. Sumit, S. B., Joshi, S. & Rana, U. Comprehensive review of r-CNN and its variant architectures.
42. Zhou, T., Li, Z. & Zhang, C. Enhance the recognition ability to occlusions and small objects with robust faster r-CNN. *Int. J. Mach. Learn. Cybern.* **10**, 3155–3166 (2019).
43. Khanam, R. & Hussain, M. Yolov11: an overview of the key architectural enhancements. *Arxiv Preprint Arxiv*, 2410. https://doi.org/10.48550/arXiv.2410.17725 (2024).
44. Casas, E., Ramos, L., Bendek, E. & Rivas-Echeverría, F. Assessing the effectiveness of YOLO architectures for smoke and wildfire detection. *IEEE Access.* **11**, 96554–96583 (2023).
45. Bakirci, M. & Bayraktar, I. Harnessing UAV technology and YOLOv9 algorithm for real-time forest fire detection. 2024 International Russian Automation Conference (RusAutoCon). IEEE. pp. 95–100. (2024).
46. Saydirasulovich, S. N., Mukhiddinov, M., Djuraev, O., Abdusalomov, A. & Cho, Y. An improved wildfire smoke detection based on YOLOv8 and UAV images. *Sens. (Basel)*. **23** (20), 8374 (2023).
47. Alkhammash, E. H. A comparative analysis of YOLOv9, YOLOv10, YOLOv11 for smoke and fire detection. *Fire 8*(1), 2571–6255 (2025).
48. Glenn Jocher, J. Q., Ultralyrics & YOLO11. Available*form: Https://Github.Com/Ultralytics/Ultralytics.[Accessed 10.26 2024]. (2024).
49. Awad, A. & Aly, S. A. Early diagnosis of acute lymphoblastic leukemia using YOLOv8 and YOLOv11 deep learning models. *arXiv preprint* arXiv:2410.10701 (2024).
50. Ultralytics Ultralytics yolov11. Available*form: (2024). https://docs.ultralytics.com/models/yolo11/s [Accessed 10.24 2024].
51. Sharma, A., Kumar, V. & Longchamps, L. Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and faster r-CNN models for detection of multiple weed species. *Smart Agricultural Technol.* **9**, 100648 (2024).
52. Ren, S., He, K., Girshick, R. & Sun, J. Faster r-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39** (6), 1137–1149 (2016).
53. Tan, M., Pang, R. & Le, Q. V. Efficientdet: scalable and efficient object detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10781-90. (2020).
54. Lin, T., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision. pp. 2980-8. (2017).
55. Liu, W. et al. 14th European Ssd: single shot multibox detector. Computer Vision–ECCV, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer. 2016. pp. 21–37. (2016).
56. Carion, N. et al. End-to-end object detection with transformers. European conference on computer vision. Springer. pp. 213 – 29. (2020).
57. Cai, Z. & Vasconcelos, N. Cascade r-cnn: delving into high quality object detection. Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6154-62. (2018).
58. Xu, Z., Hrustic, E. & Vivet, D. Centernet heatmap propagation for real-time video object detection. Computer Vision–ECCV. : 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16. Springer. 2020. pp. 220 – 34. (2020).
59. El-Madafri, I., Peña, M. & Olmedo-Torre, N. The wildfire dataset: enhancing deep learning-based forest fire detection with a diverse evolving open-source dataset focused on data representativeness and a novel multi-task learning approach. *Forests* **14** (9), 1697 (2023).
60. Minha, A. & Forest_fire_smoke_and_non_fire_image_dataset Available*form: https://www.kaggle.com/datasets/amerzishminha/forest-fire-smoke-and-non-fire-image-dataset/data. (2023). [Accessed 2.24 2025].
61. Torralba, A., Russell, B. C. & Yuen, J. Labelme: online image annotation and applications. *Proc. IEEE Inst. Electr. Electron. Eng.* **98** (8), 1467–1484 (2010).

62. Bhattacharjee, A., Popov, A. A., Sarshar, A. & Sandu, A. Improving the adaptive moment Estimation (ADAM) stochastic optimizer through an implicit-explicit (IMEX) time-stepping approach. *Arxiv Preprint Arxiv*. **2403**, 13704 (2024).
63. Habring, A. & Holler, M. Neural-network-based regularization methods for inverse problems in imaging. *GAMM-Mitteilungen* **47** (4), e202470004 (2024).
64. Ali, U., Ismail, M. A., Habeeb, R. A. A. & Shah, S. R. A. Performance evaluation of YOLO models in plant disease detection. *J. Inf. Web Eng.* **3** (2), 199–211 (2024).
65. Correira, A., Ferreira, F. & Mišković, N. Comparing different YOLO versions for boat detection and classification in real datasets. *OCEANS 2024-Singapore* 1–4 (IEEE, 2024).
66. Coleman, A. M. *Multi-formalism Modeling for Disaster Response* (The University of Utah, 2023).
67. ON, M. H. M. JM. A comprehensive review on deep learning-based data fusion. *IEEE Access.* **12**, 180093–180124. https://doi.org/10.1109/ACCESS.2024.3508271 (2024).
68. Mowla, M. N., Asadi, D., Masum, S. & Rabie, K. Adaptive hierarchical multi-headed convolutional neural network with modified convolutional block attention for aerial forest fire detection. *IEEE Access* **13**, 12345–12355 (2024).
69. Khalil, A., Rahman, S. U., Alam, F., Ahmad, I. & Khalil, I. Fire detection using multi color space and background modeling. *Fire Technol.* **57**, 1221–1239 (2021).
70. Kaliyev, D., Shvets, O. & Györök, G. Computer vision-based fire detection using enhanced chromatic segmentation and optical flow model. *Acta Polytech. Hung.* **20** (6), 27–45 (2023).
71. Qiang, X., Zhou, G., Chen, A., Zhang, X. & Zhang, W. Forest fire smoke detection under complex backgrounds using TRPCA and TSVB. *Int. J. Wildland Fire*. **30** (5), 329–350 (2021).
72. Wang, H., Bai, Y. & Li, H. YOLOV5s-fire: a lightweight model for flame detection. Proceedings of the 2023 6th International Conference on Image and Graphics Processing. pp. 51 – 7. (2023).
73. Mambile, C., Kaijage, S. & Leo, J. Application of deep learning in forest fire prediction: a systematic review. *IEEE Access* **12**, 5678–5689 (2024).
74. Khan, S. M. et al. A systematic review of disaster management systems: approaches, challenges, and future directions. *Land. (Basel).* **12** (8), 1514 (2023).
75. Mowla, M. N., Asadi, D., Tekeoglu, K. N., Masum, S. & Rabie, K. UAVs-FFDB: a high-resolution dataset for advancing forest fire detection and monitoring using unmanned aerial vehicles (UAVs). *Data Brief.* ;**55**:110706. https://doi.org/10.1016/j.dib.2024.110706. (2024).
76. Mowla, M. N., Mowla, N., Shah, A. S., Rabie, K. M. & Shongwe, T. Internet of things and wireless sensor networks for smart agriculture applications: a survey. *IEEE Access.* **11**, 145813–145852 (2023).
77. Ma, J. et al. Advances in geochemical monitoring technologies for CO2 geological storage. *Sustainability* **16** (16), 6784. https://doi.org/10.3390/su16166784 (2024).
78. Ma, J., Zhou, Y., Cao, W. & Jinfeng, L. Research on a CO2 internet of things online monitoring system for geotechnical engineering construction.; (2024).

Author contributions

Conceptualization, L.H.; methodology, L.H.; software, L.H.; validation, L.L., Y.Z. and J.M.; writ-ing—review and editing, J.M.; visualization, J.M.; supervision, Y.Z.; project administration, Y.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the man-uscript.

Funding

This research was supported by the National Key Research and Development Plan (2022YFF0801201), the National Natural Science Foundation of China (U1911202), the Guangdong Key Areas Research and Development Project (2020B1111370001).

Declarations

Competing interests

The authors declare no competing interests.

Conflict of interest

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Y.Z. or J.M.