Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Dynamic Lightweight YOLO for UAV-Based Forest Fire Detection\

Xiang Yin
Wuhan Institute of Technology
Yinxia Xu
xuyinxia@wit.edu.cn
Wuhan Institute of Technology
Ruofan Zhang
Wuhan Institute of Technology

Research Article

Keywords: Smoke and Flame Detection, Deep Learning, YOLOv8, Object Detection

Posted Date: August 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-7892412/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Dynamic Lightweight YOLO for UAV-Based Forest Fire Detection

1st Xiang Yin
Computer Science and Engineering
Wuhan Institute of Technology
Wuhan, China
xiangyin@stu.wit.edu.cn

2nd Yinxia Xu*
Computer Science and Engineering
Wuhan Institute of Technology
Wuhan, China
xuyinxia@wit.edu.cn
*Corresponding author

3rd Ruofan Zhang
Computer Science and Engineering
Wuhan Institute of Technology
Wuhan, China
2692156805@qq.com

***Abstract*—Unmanned aerial vehicle (UAV) forest fire detection faces persistent challenges, including low accuracy for small smoke and flame targets, high missed and false detection rates, and the need for lightweight models due to limited onboard computational resources. This study presents an improved lightweight object detection algorithm based on YOLOv8n. First, a dynamic convolution module is integrated into the backbone to optimize resource allocation, enhancing feature extraction while maintaining low FLOPs and reducing false detections in complex backgrounds. Second, a dynamic detection mechanism with scale, spatial, and task awareness is employed in the detection head to improve recognition of multiscale objects, especially small-scale smoke and flames. Third, a cross-spatial-learning-based attention module is introduced into the feature fusion neck to recalibrate channel weights and strengthen multiscale feature representation. Finally, depthwise separable convolution combined with structural reparameterization is applied to further lighten the neck network. Experimental results on a public dataset show that the proposed model achieves a mean average precision (mAP) of 69.5% with only 10.6 MB in size, surpassing the baseline YOLOv8n by 2.7% mAP and 4.2% accuracy. These improvements significantly enhance detection performance while maintaining a compact design, making the approach well-suited for real-time UAV-based forest fire monitoring.**

***Index Terms*—Smoke and Flame Detection; Deep Learning; YOLOv8; Object Detection;**

I. Introduction

In recent years, the frequent occurrence of forest fires has caused significant losses to human life and property, posing a severe threat to the social and ecological environment [1], [2]. Therefore, timely fire warnings and effective fire control are crucial. Traditional forest fire detection methods mainly include manual patrols [3], ground-based monitoring [4], and satellite remote sensing technology [5]. Manual patrols can provide relatively accurate on-site fire information, but they have limited coverage, low efficiency, and are easily affected by terrain and weather conditions, leading to long response times [6]. Ground-based monitoring, which relies on fixed monitoring stations or sensor networks, can collect real-time environmental data such as temperature and humidity with high accuracy and fast response times [7]. However, its monitoring range is limited and cannot effectively handle large-scale fires. Satellite remote sensing technology utilizes high-altitude imaging to detect large-scale fire hotspots, offering a clear coverage advantage and the ability to update data rapidly [8]. But, its accuracy is affected by resolution limitations and weather interference, and its response time is relatively long [9].

With the rapid development of UAV technology and computer vision, fire detection based on UAV imagery has gradually become an essential means of forest fire monitoring [10], [11]. This technology can effectively adapt to complex terrains, capture fire dynamics in real time, and has relatively low deployment costs. Traditional image-based fire detection methods typically rely on features such as color, motion, shape, and texture to identify fire targets. However, these methods suffer from long computation times and high subjectivity, making it difficult to achieve high-precision detection in complex environments. In contrast, deep learning-based methods can effectively overcome the limitations of traditional approaches, significantly improving detection accuracy and efficiency. Moreover, they offer stronger generalization capabilities and adaptability, enabling stable fire feature recognition in complex and dynamic natural environments, leading to remarkable advancements in the field of object detection.

In the field of deep learning-based object detection, algorithms are generally categorized into two types [12]. The first type consists of single-stage detection algorithms, including SSD [13] and the YOLO series. The second type includes two-stage detection algorithms, with representative models such as R-CNN and Faster R-CNN [14]. Although higher versions of YOLO exist, such as YOLOv9, YOLOv10, and YOLOv11, the YOLOv8 algorithm stands out for its high accuracy, relatively low computational cost, and fast processing speed, making it particularly suitable for real-time object detection tasks. When operating without a GPU or with a lightweight GPU, it offers exceptional cost-effectiveness. Compared to YOLOv10 and YOLOv11, YOLOv8 is more suitable for deployment on mobile devices and performs better in detecting small objects, while also requiring lower hardware costs. YOLOv8 has several architectural variants, including YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x, which differ in network depth and feature map width, with YOLOv8n having the smallest parameter size and computational complexity [15], [16]. The YOLOv8 framework primarily consists of three

components: the backbone network, the neck network, and the head network. The backbone network is responsible for feature extraction, the neck network enhances the integration of features provided by the backbone to improve target localization capability, and the head network utilizes the extracted features for final predictions.

For fire detection, Ilyas et al [17]. combined SVM classification with a deep CNN architecture to enhance the early prediction capability and accuracy of forest fire detection, effectively reducing the false alarm rate. Wang [18] adopted an improved SSD algorithm, significantly improving the detection capability for low-resolution images and small targets. Additionally, some studies have employed the Faster R-CNN algorithm to distinguish fire and non-fire regions, integrating Grassmann manifold analysis or long short-term memory (LSTM) methods to classify features within bounding boxes, further improving fire detection accuracy. Yang [19] and Xue [20] et al. introduced SCConv and CBAM attention modules to enhance YOLOv8 and YOLOv5, respectively, improving the model's adaptability to complex forest fire scenarios, particularly excelling in small fire target detection. Furthermore, Xiao et al [21]. incorporated the SimAm attention mechanism into YOLOv7, further emphasizing fire target features and improving detection performance.

Although deep learning-based fire detection methods have made significant progress, many challenges remain.As shown in Fig. 1, detecting small targets in complex environments is particularly difficult, as flames often exhibit blurred boundaries, leading to reduced recognition and detection accuracy [22]. Moreover, environmental complexity and variations in shooting angles increase the difficulty of model adaptation. Due to the hardware limitations of UAVs, the model's size and computational capacity are constrained, further affecting detection precision and reliability. Existing fire detection algorithms still have shortcomings in accuracy and real-time performance, especially when dealing with low-resolution images and complex scenes, where detection effectiveness is severely impacted. The diverse shapes and colors of flames, along with influences from weather conditions, lighting variations, and background interference, often lead to false detections and missed detections in deep learning models [23]. Additionally, the detection process is time-consuming, making it difficult to meet the requirements for real-time monitoring.

To address the above issues, this paper proposes an improved fire detection algorithm based on the YOLOv8n model, aiming to enhance detection performance under UAV hardware constraints. First, a dynamic convolution module [24] is introduced into the backbone network to dynamically optimize computational resource allocation, significantly improving network performance under low FLOPs conditions. Second, the DyHead module [25] is integrated into the detection head, enhancing the model's perception of object size, spatial information, and task characteristics, enabling it to flexibly adjust detection strategies according to different target sizes, thereby improving detection accuracy. Next, an efficient multi-scale attention mechanism based on cross-spatial learning [26] is incorporated into the network's neck structure. By encoding global information, it recalibrates and strengthens the channel weights in each parallel branch, further enhancing feature representation capability. Finally, the Slim-neck module [27] is introduced to optimize the convolutional neural network's neck structure, reducing model size while further improving computational efficiency.



Fig. 1: Challenges in Forest Fire Detection: The first row shows small - target flame scenarios. The second row presents complex backgrounds and shooting angle variations. The third row displays diverse weather and lighting change conditions.

II. YOLOv8 Network Algorithm

In forest fire detection tasks, various factors such as weather, lighting, and background conditions can significantly impact detection performance. The original YOLOv8n model suffers from low detection accuracy and a high false detection rate. To address these issues, this paper introduces the following improvements:

(1) Integration of the Dynamic Convolution module: The conventional convolution layers in the backbone network are replaced with DynamicConv, enabling dynamic allocation and optimization of computational resources, effectively enhancing network performance under low FLOPs conditions.

(2) Incorporation of the DyHead module: This module is added to the detection head to improve the model's perception of object size, spatial information, and task characteristics, allowing it to adaptively adjust detection strategies based on different target sizes, thereby significantly enhancing detection accuracy.

(3) Implementation of a multi-scale attention mechanism based on cross-spatial learning: This mechanism is introduced into the network's neck structure to recalibrate the channel weights of each parallel branch through global information encoding, thereby strengthening feature representation.

(4) Adoption of the Slim-neck module: The convolutional neural network's neck structure is optimized to reduce model

parameters and size while maintaining high performance. The structure of the improved network model is shown in Fig. 2.

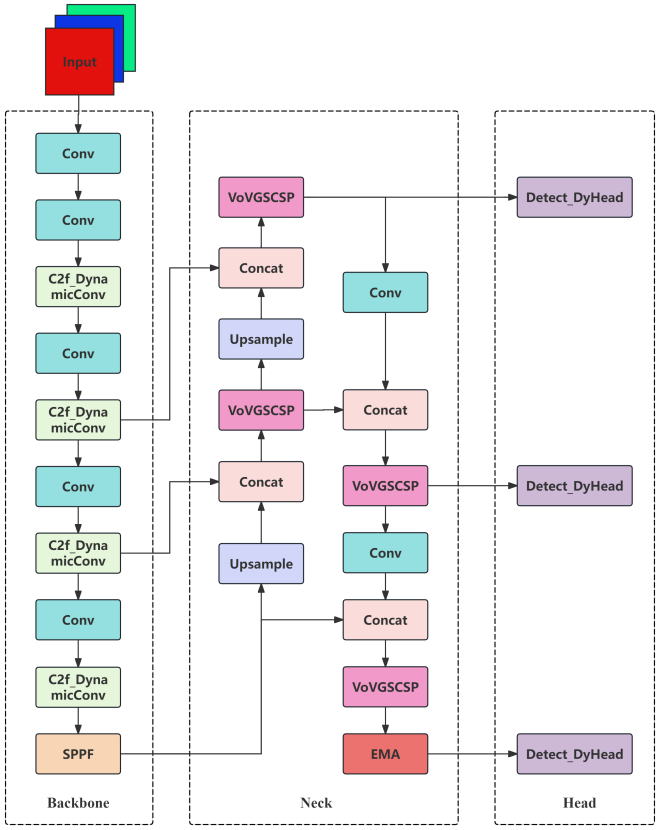


Fig. 2: The improved YOLOv8n model structure.

A. Dynamic Convolution Backbone

In forest fire detection tasks, the complex and ever-changing terrain of forests, along with unstable lighting conditions, often leads to uneven brightness distribution in images. This reduces the contrast between fire targets and the background, posing significant challenges to detection. Meanwhile, smoke and flames exhibit high dynamism and irregular shapes, rapidly moving and deforming in the air, causing their edges and contours to become blurred, which further affects detection accuracy. Additionally, detection scenes are often highly complex, containing various background elements such as buildings, roads, and vegetation. These redundant details and interference factors make it difficult to accurately distinguish fire and smoke targets from the complex background. However, the dynamic convolution module demonstrates unique advantages in addressing these challenges. By dynamically selecting or combining different convolution kernels for each input sample, dynamic convolution allows the network to adaptively adjust its parameters based on variations in the input data. This adaptability enables the model to better handle the complexities of forest fire detection tasks, significantly improving detection accuracy and reliability [28].

To enhance the model's adaptability to different input features, we introduce a dynamic convolution module. The dynamic convolution adjusts its parameters dynamically based on the input features, allowing the model to adapt to changes in the input data and thereby enhancing its representational power. The core idea behind dynamic convolution is to adjust the convolution kernel based on the input features dynamically, which can be implemented through the following formula:

$$y = g(\tilde{W}(x)x + \tilde{b}(x)) \tag{1}$$

Here, y is the output of the model, g is the activation function, $\tilde{W}(x)$ and $\tilde{b}(x)$ are the dynamically generated weight matrix and bias, respectively, which are adjusted dynamically based on the input features x. The dynamic adjustment of the weight matrix and bias is achieved through the following methods:

$$\begin{aligned} \tilde{W}(x) &= \sum_{k=1}^{K} \pi_k(x)\tilde{W}_k, \\ \tilde{b}(x) &= \sum_{k=1}^{K} \pi_k(x)\tilde{b}_k, \end{aligned} \tag{2}$$

Here, K is the total number of convolution kernels, $\tilde{W}_k$ and $\tilde{b}_k$ are the weights and biases of the k-th convolution kernel, respectively, and $\pi_k(x)$ is the attention weight used to dynamically select or combine different convolution kernels. The attention weights satisfy the following conditions:

$$0 \leq \pi_k(x) \leq 1, \quad \sum_{k=1}^{K} \pi_k(x) = 1 \tag{3}$$

These conditions ensure that the attention weights can be interpreted as probability distributions, thereby making the combination of convolution kernels a weighted average. The advantage of the dynamic convolution module lies in its adaptability and flexibility, allowing the model to adjust its parameters dynamically according to different input features, thus enabling the model to better adapt to different data distributions and task requirements.

The structure of the dynamic convolution module (Fig. 3) adopts multiple convolution kernels or parameter sets and dynamically selects the appropriate convolution kernel according to input features.First, the input data x is simultaneously fed into multiple parallel convolution operations $(\text{conv}_1, \text{conv}_2, ..., \text{conv}_k)$ and an attention module. In the attention module, the input data x first undergoes global average pooling to extract global spatial information. Then, it passes through two fully connected layers, with a ReLU activation function in between. Finally, a Softmax function is applied to obtain the normalized attention weights π_k.These attention weights π_k are multiplied by their corresponding convolution kernels, and the weighted convolution kernels are aggregated to form the final aggregated kernel. This aggregated kernel is then convolved with the input data x, followed by batch normalization and an activation function, generating the final

output y.This process dynamically aggregates multiple convolution kernels through the attention mechanism, enhancing the model's representation capability while improving performance without increasing the network's depth or width.

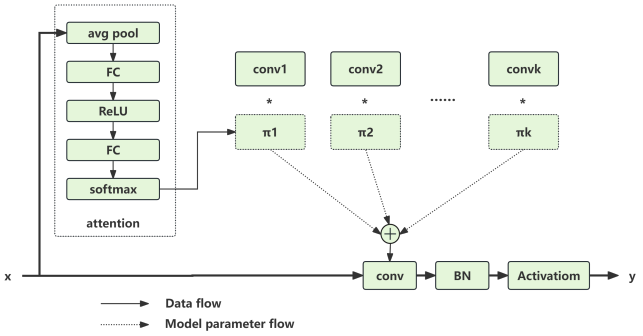


Fig. 3: Dynamic Convolution Module.

Dynamic convolution maintains high parameter efficiency by sharing and dynamically combining convolution kernels, significantly reducing the number of model parameters while only slightly increasing computational cost. This approach not only improves the model's generalization ability but also enhances its adaptability, allowing it to dynamically adjust kernel selection based on different input features. Dynamic convolution enables the deployment of more complex network structures in resource-constrained environments without significantly increasing computational burden. By dynamically adjusting and optimizing the utilization of computing resources, dynamic convolution achieves performance improvement under low FLOPs conditions, making it particularly suitable for running efficient models on devices with limited computing resources.

B. Multi-dimensional Dynamic Detection Head

In object detection models, the detection head is responsible for classifying and regressing feature maps, and its performance directly affects the accuracy of object localization and recognition. However, in forest fire detection scenarios, the dynamic nature of flame features, irregular shapes, and similarities with the background pose significant challenges to traditional detection heads. To enhance the model's detection capability in complex environments, this study adopts the DyHead (Dynamic Head) module to replace the original detection head in YOLOv8n, effectively improving the fire detection model [29].

The DyHead module dynamically adjusts the positional, spatial, and channel information of feature maps, providing more precise feature representation for fire detection tasks. The structure of the DyHead module is shown in Fig. 4, and its feature processing is divided into three core stages: scale-aware attention (π_L), spatial-aware attention (π_S), and task-aware attention (π_C. The core operation of the Dynamic Head framework is described by Equation (4), which illustrates the

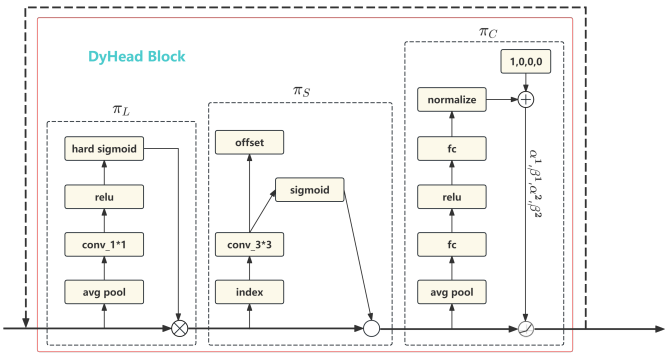


Fig. 4: Dynamic Head Block.

enhancement process of multi-dimensional attention mechanisms on input features:

$$F_{\text{out}} = \pi_L(F) \times \pi_S(F) \times \pi_C(F) \times F \tag{4}$$

Scale-Aware Attention (π_L): In the dynamic position weighting module, DyHead uses a hard Sigmoid function, 1×1 convolution, and global average pooling to dynamically model the positional information of feature maps. This module enhances the model's ability to perceive object edge features, making it more adaptable to irregular flame boundaries. In fire detection scenarios, the edges of flames and smoke are often crucial indicators, and the introduction of the dynamic position weighting module significantly improves the model's localization accuracy for these regions.

Spatial-Aware Attention (π_S): This module optimizes the spatial relationships of objects in fire detection scenes through convolution operations and spatial offset modeling. Flames and smoke often exhibit a diffusive or dispersed state in space, and the π_S module captures these variations across different spatial dimensions. By accurately modeling spatial features, this module effectively enhances the model's adaptability to fire detection, particularly in complex forest backgrounds, where it can better distinguish between flames, smoke, and other interfering elements such as leaves or sunlight reflections.

Task-Aware Attention (π_C): This module dynamically adjusts feature map weight distribution across channels through global normalization and fully connected layers. In forest fire scenarios, variations in flame and smoke color, morphology, and dynamic lighting conditions can lead to uneven feature distribution across channels, making it difficult for traditional detection heads to comprehensively capture these variations. The π_C module learns global context information, adaptively enhances the representation of key channels, and suppresses background interference, thereby allowing the model to focus more on high-brightness flame regions and the diffusion characteristics of smoke.

In forest fire detection tasks, the DyHead module significantly improves the detection performance of the YOLOv8 model by processing dynamic position, spatial, and channel information in a coordinated manner. This module accurately

captures dynamic changes in fire scenes, such as rapidly moving jumping flames, enabling high-precision detection in complex backgrounds. Additionally, the DyHead module effectively enhances the model's ability to detect small flames and early-stage smoke, while also significantly reducing the false detection rate.

Compared with traditional detection heads, the DyHead module dynamically adjusts multi-dimensional information in feature maps, providing a more efficient feature representation method for fire detection models. This effectively addresses the core challenges in forest fire detection, offering comprehensive support for improving model performance.

C. Efficient Multi-scale Attention

In the YOLOv8n model, the backbone network does not include specialized modules for distinguishing different objects. Traditional attention mechanisms such as SENet and CBAM compress channel information using fully connected layers, where global average pooling (GAP) only generates channel-level statistics without preserving spatial position correlations. Additionally, single-scale pooling operations struggle to adapt to the multi-scale characteristics of fire. To address these limitations, this study introduces an Efficient Multi-Scale Attention (EMA) mechanism based on cross-spatial learning, which significantly improves the model's ability to capture fire-related features through global encoding, dynamic channel weight adjustment, and multi-scale feature fusion [30].

This paper incorporates the EMA attention mechanism into the neck structure of the YOLOv8 model. The complete structure of the EMA mechanism is illustrated in Fig. 5. First, the EMA mechanism encodes the input feature maps globally, extracting global contextual information across both the channel and spatial dimensions to optimize fire feature representation. In fire scenarios, flame highlights and smoke edges often exhibit different characteristics at varying scales. EMA enhances the saliency of these key targets by modeling cross-scale correlations. At the same time, the EMA mechanism dynamically adjusts the weight of each channel during feature fusion, allowing the model to focus more on critical objects such as flames and smoke while suppressing irrelevant background information. Through its cross-spatial multi-scale feature fusion strategy, the EMA mechanism extracts multi-dimensional information of fire across different scales, significantly enhancing the model's feature representation capability and robustness, particularly for small target detection.

Figure 5 illustrates the structure of the EMA mechanism. By enhancing the saliency of flame highlights and smoke edges, EMA effectively reduces false detections. The variability and complexity of fire, especially the dynamic nature of flames and smoke under poor lighting conditions or adverse weather (such as thick fog or nighttime settings), often challenge detection accuracy. By modeling global contextual information, the EMA mechanism improves the model's adaptability to such complex environments, ensuring high detection accuracy even under suboptimal conditions. The EMA attention mechanism optimizes feature fusion, strengthens the expression of key

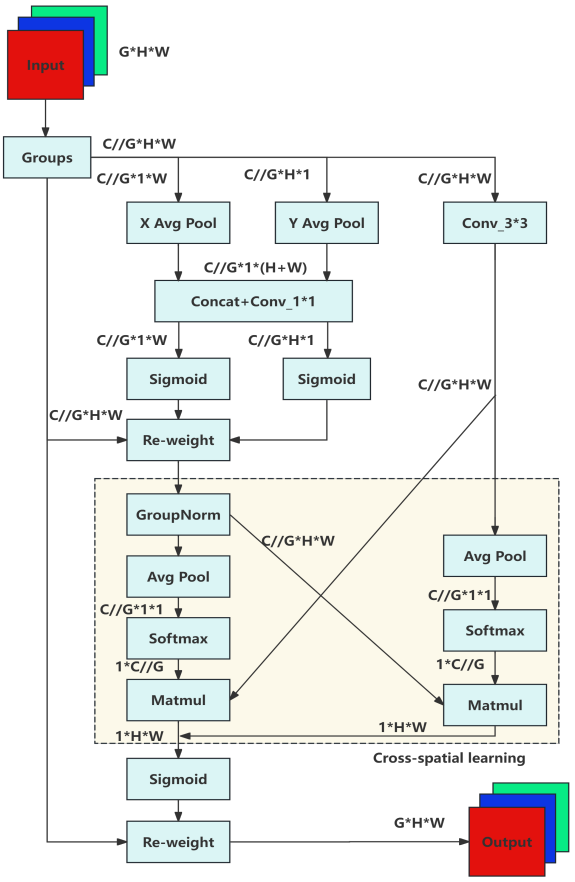


Fig. 5: Efficient Multi-Scale Attention Module

target regions, and significantly improves the performance of YOLOv8 in forest fire detection tasks. Notably, in complex wildfire scenarios, EMA effectively reduces both missed and false detections, enhances small target detection capabilities, and improves adaptability to dynamic changes and adverse environmental conditions. This provides reliable technical support for UAV-based fire monitoring. Furthermore, EMA enhances the detection of small flames and early-stage smoke, significantly improving the model's ability to recognize early fire signals and effectively reducing missed detection rates. This mechanism offers a more efficient and reliable solution for forest fire detection tasks.

D. Lightweight Feature Fusion Neck Structure

In forest fire detection tasks, the complex and variable forest terrain, along with unstable lighting conditions, often leads to uneven brightness distribution in images. This reduces the contrast between fire targets and the background, making detection more challenging. Additionally, smoke and flames exhibit highly dynamic and irregular shapes, with blurred edges and contours, further affecting detection accuracy. The detection scene is also highly complex, containing various background elements such as buildings, roads, and vegetation, which interfere with accurate fire detection. To address these

challenges, this study integrates the Slim-neck module into the neck structure of the YOLOv8 model.

The Slim-neck module is a lightweight design specifically for object detection model neck structures, aiming to reduce computational complexity while improving feature fusion efficiency and representation capability. Its core structure is based on VoV-GSCSP, where an initial convolution is used to adjust the input feature channels, followed by two groups of GSConv modules for efficient processing. The computational formula is as follows:

$$\text{Output} = \text{Shuffle}(\text{Concat}(\text{Conv}(X), \text{DWConv}(X))) \quad (5)$$

where DWConv represents depthwise convolution, Conv represents standard convolution, and Concat denotes concatenation operation.

The GSConv module structure is shown in Fig. 6. GSConv combines group convolution and channel shuffle operations, which effectively reduces computational costs while enhancing inter-channel information exchange. The processed features are fused through concatenation operations and finally transformed via a convolution layer, ensuring that the extracted features retain both global information and detailed feature representation. This multi-branch design allows the Slim-neck module to significantly enhance feature diversity and expressiveness while maintaining a lightweight architecture.

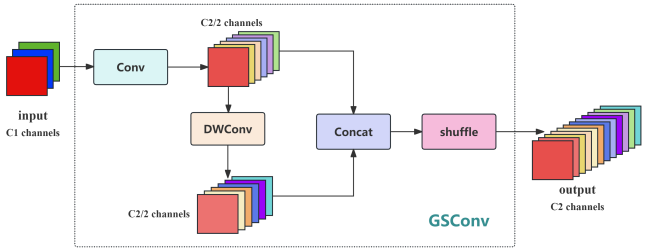


Fig. 6: GSConv Module Structure Diagram

Traditional neck structures, such as FPN and PAN, often rely on complex feature fusion methods, resulting in high computational costs, making it difficult to apply them directly to resource-constrained platforms. The Slim-neck module optimizes the design, significantly reducing parameter size and floating-point operations FLOPs, making the object detection model more suitable for deployment under limited computational resources. The GS bottleneck module and VoV-GSCSP module structures are illustrated in Fig. 7. The multi-branch design of VoV-GSCSP enhances feature diversity while improving the model's capability in complex fire detection scenarios. When facing the dynamic changes of forest fires and background interference, the Slim-neck module exhibits excellent robustness, providing an effective solution for early fire detection, particularly in addressing small target detection and insufficient feature extraction problems.

By integrating the Slim-neck module, the model achieves a better balance between detection accuracy and inference

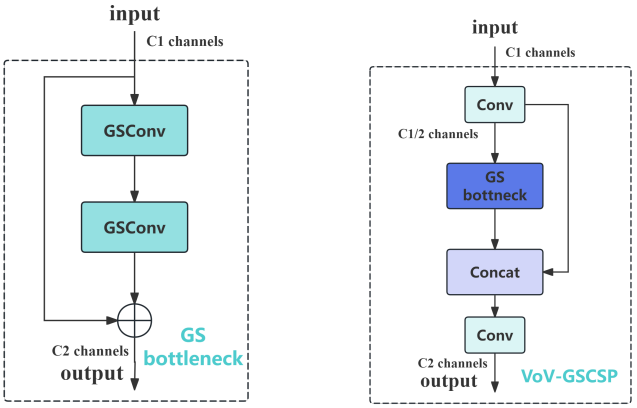


Fig. 7: Two other Slim-neck modules.(**a**) GS bottleneck module. (**b**) VoV-GSCSP module.

speed. After deployment on a UAV platform, the model can perform real-time fire monitoring under low power consumption conditions. Particularly in forest fire detection scenarios, the Slim-neck module's optimization enhances sensitivity to weak features, such as small flames and early-stage smoke, providing strong technical support for early fire warning systems. This lightweight yet efficient design not only overcomes the bottleneck of traditional neck structures but also opens up new possibilities for fire detection applications in resource-constrained environments.

III. Results

A. Dataset and Experimental Environment

We conducted comparative experiments on the publicly available dataset C4-AI (the self-built fire and smoke dataset from the 2023 China Collegiate Computing Contest - Artificial Intelligence Creative Competition). The dataset contains 9,848 images, primarily categorized into two classes: Fire and Smoke. It is divided into training, validation, and test sets in a 7:2:1 ratio.

The GPU used in the experiment is the NVIDIA RTX4060-8GB, with a 16-core CPU. The PyTorch version is 2.0.0, Python version is 3.9.0, and CUDA version is 11.7. The specific experimental parameters are shown in Tab. I.

TABLE I: Training Parameter Settings

Parameter	Value
epochs	150
patience	20
batch size	16
workers	8
optimizer	SGD
image size	640
close_mosaic	10
initial lr (lr0)	0.01
final lr (lrf)	0.01

The input image size for all models in the experiment is 640×640. Under the same experimental conditions, all models are trained for 150 epochs using the same training parameters. If no performance improvement is observed during the last 20 epochs, the early stopping mechanism is triggered to end the training early. All YOLO models we used were trained on this public dataset with the settings shown in Table 1.

B. Evaluation Metrics

Precision is an important metric for evaluating the performance of object detection models, especially critical in forest fire detection tasks. Precision represents the proportion of actual fire targets among the predicted fire targets, reflecting the model's ability to reduce false positives.TP represents the number of samples where both the true class and the model's predicted class are positive, while FP represents the number of samples where the true class is negative but the model predicts it as positive. The formula for precision is shown in Eq. 6:

$$P = \frac{TP}{TP + FP} \tag{6}$$

Average precision is a performance metric that measures the accuracy of predicted bounding boxes and classes. P represents precision, and R represents the ratio of correctly predicted samples to the total actual samples. The integral calculation results in the area enclosed by the P-R curve and the first quadrant. The formula for average precision is shown in Eq. 7:

$$AP = \int_0^1 P(R)dR \tag{7}$$

mAP is a core evaluation metric in the field of object detection, used to comprehensively assess the model's detection performance across multiple categories. The calculation method involves first computing the average precision (AP) for each category based on the overlap between the predicted and ground truth boxes, using IoU as a threshold. Then, the mean of the AP values for all categories is calculated. The formula for the mean average precision is shown in Eq. 8:

$$mAP = \frac{1}{n} \sum_{i=1}^{n} AP_i \tag{8}$$

mAP50 represents the average precision at an IoU threshold of 0.5, while mAP50-95 considers multiple IoU thresholds ranging from 0.5 to 0.95, providing a more comprehensive reflection of the model's detection performance. mAP50-95 calculates the average precision across multiple IoU thresholds, offering a more holistic evaluation of the model's performance, especially in scenarios requiring precise localization.

Frames per second (FPS) reflects the real-time processing capability of the model and is a key performance indicator for fire detection algorithms on drone platforms. FPS indicates the number of image frames the model can process per second, which is critical for real-time fire monitoring. High FPS ensures that drones can quickly capture fire signals while monitoring large forest areas and provide real-time early warning capabilities. The main factors affecting FPS include the model's computational complexity, hardware performance, and input image resolution. In practical applications, to balance detection accuracy and real-time performance, it is necessary to optimize the model architecture and reduce computation to improve FPS, meeting the requirements for real-time fire monitoring.

The number of model parameters and floating-point operations FLOPs are important indicators of the complexity of deep learning models and are significant for algorithm deployment on drones. The model parameters represent the number of trainable parameters, usually measured in millions , while FLOPs represent the number of floating-point operations required during a forward pass of the model. Smaller model parameters and FLOPs indicate a more lightweight model with higher computational efficiency, making it more suitable for resource-constrained drone devices. For forest fire detection tasks, choosing models with lower parameters and FLOPs can reduce hardware costs and improve drone battery life, providing technical support for large-scale and long-duration fire monitoring.

Model size refers to the storage requirements of the model, typically measured in megabytes(MB). It directly affects the storage usage and deployment efficiency of drone devices. In fire detection tasks, drones are often limited by storage space and transmission speed, making model size optimization particularly important. By adopting techniques such as quantization and pruning, the model can significantly reduce in size while maintaining detection accuracy, thereby reducing storage and communication burdens, and further improving system response speed and practicality.

C. Comparative Experiment Analysis

To verify the effectiveness of the proposed improved algorithm, we made enhancements based on the YOLOv8n model and conducted comparative experiments with other mainstream object detection algorithms on a public forest fire dataset. The experiments primarily compare model performance across multiple dimensions, including precision (P), recall (R), mean average precision at an IoU threshold of 0.5 (mAP50), mean average precision across multiple IoU thresholds (mAP50-95), the number of model parameters (Params), computational complexity and model size. A comprehensive analysis is conducted to evaluate the applicability and advantages of each model in forest fire detection tasks. The table presents the experimental results of YOLOv5n, YOLOv7t, YOLOv8n, and other models, along with the performance of our improved model. The comparative results are shown in Tab. II.

In addition, we compare the changes in average precision during the training process of different YOLO series models and our improved model to illustrate the dynamic performance variations, as shown in Fig. 8 and Fig. 9.

Our improved model has achieved significant advantages in two key metrics: precision and mAP50. In terms of detection

TABLE II: Comparison of Experimental Results for Different Models

Model	P/%	mAP50/%	mAP50-95/%	Params/M	FLOPs/G	Model Size/MB
YOLOv5n	62.2	59.5	28.7	**1.9**	**4.5**	7.4
YOLOv7t	62.9	64.3	31.4	6.0	13.2	11.7
YOLOv8n	70.6	66.8	38.0	3.0	8.2	6.3
YOLOv9t	70.6	66.7	38.6	2.0	7.7	23.4
YOLOv10n	71.8	64.2	35.8	2.7	8.4	10.9
YOLOv11n	71	58.6	30.4	2.6	6.5	**5.4**
Ours	**75.7**	**69.5**	**39.3**	5.4	13.0	10.6

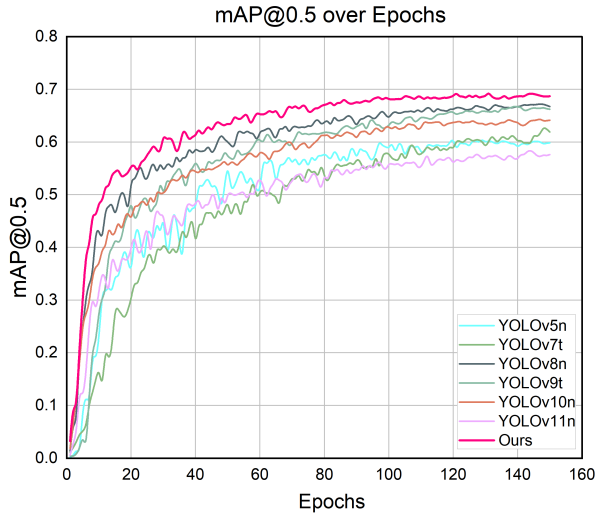


Fig. 8: Comparison of mAP@0.5 Across Different Models

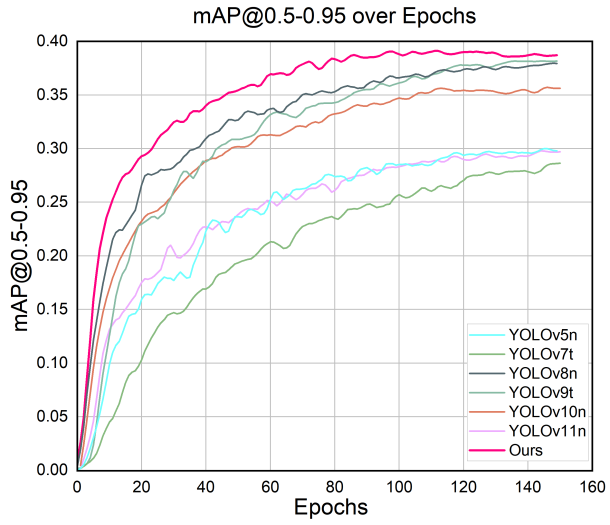


Fig. 9: Comparison of mAP@0.5-0.95 Across Different Models

performance, compared to the baseline model YOLOv8n, precision has increased by 5.1%, and mAP50 has improved by 2.7%, demonstrating excellent small-object detection capabilities. Compared to YOLOv7t, which has a larger number of parameters, our model achieves a 7.9% improvement in mAP50-95 while reducing the number of parameters from 6.0M to 5.4M, indicating that it effectively controls model size while enhancing detection accuracy. In terms of computational complexity and model size, the improved model maintains a lightweight design, with a weight size of 10.6MB and a floating-point computation volume of 13.0G, making it suitable for deployment on resource-constrained drone devices.

From the comparison with other models, it can be observed that while YOLOv9T achieves mAP50 and mAP50-95 scores close to YOLOv8n, its recall and precision are lower, making it difficult to meet the high requirements of forest fire detection. Meanwhile, YOLOv5n and YOLOv11 are inferior to our method in overall performance due to limitations in model parameters or computational complexity. The improved model enhances performance under low-FLOPs conditions through the Dynamic Convolution (DynamicConv) module, integrates the DyHead module to strengthen the multi-task perception capability of the detection head, and further optimizes feature representation and model structure by incorporating the EMA attention mechanism and Slim-neck structure, ultimately achieving a well-balanced performance.

Additionally, we conducted comparative experiments using the trained model weights to predict sample images, with experimental results shown in Fig. 10. Compared to the original YOLOv8n model, the improved model demonstrates higher detection accuracy, stronger multi-target detection capabilities, and fewer false detections in fire detection tasks. In flame region recognition, the detection bounding boxes of the improved model better fit the actual shape and range of the flames, enabling more precise localization of fire locations. This is particularly evident in complex scenarios, such as burning trees in forests and scenes where firefighters are present, where the detection boxes tightly enclose the flames, avoiding coverage of non-fire areas. Furthermore, the improved model is more sensitive to detecting small flame targets, accurately identifying and marking each fire region in scenes with multiple fire points without missing or merging targets. At the same time, it reduces false detections in non-fire areas, improving the reliability of detection results and providing more accurate information support for fire monitoring and rescue operations.



Fig. 10: Comparison of Experimental Prediction Results

D. Ablation Experiment

In this study, we conducted ablation experiments on the UAV-based forest fire detection algorithm based on the YOLOv8n model to evaluate the impact of different attention mechanisms on model performance. By comparing the baseline YOLOv8n model with models incorporating different attention mechanisms (including CBAM, ECA, CA, GAM, and EMA), we quantified the specific effects of each mechanism on model accuracy, computational efficiency, and model size. The ablation study on attention mechanisms is shown in Tab. III.

The application of the EMA mechanism significantly improved the model's performance. According to the results of the ablation experiment on attention mechanisms, the model with EMA achieved an accuracy of 72.5% and a mAP50 of 67.5%, both of which are higher than those of other attention mechanisms and the baseline model. This indicates that EMA plays a key role in enhancing the model's detection accuracy for small targets in forest fires.

The parameter count (Params/M) and computational complexity (FLOPs/G) of the EMA model are similar to those of the smallest models and can be considered negligible. Although the model size is 11.9MB, it does not affect deployment on drones. Additionally, the EMA model's frame rate (FPS/f*s-1) reached 498.97, the highest among all models, demonstrating its speed advantage. EMA's efficient multi-scale attention capability makes it a critical technology for improving detection accuracy, with significant practical value and relevance for UAV-based forest fire detection tasks. To verify the impact of different improvement strategies on the initial model and the coupling between different modules, we performed ablation experiments on our improvement strategies, as shown in Tab. IV.

The experimental results show that the individual introduction of C2f_DynamicConv and EMA significantly improved the model's accuracy and mean average precision. C2f_DynamicConv enhanced the model's dynamic feature extraction capability, improving its adaptability to complex scenes. EMA, on the other hand, introduced a multi-scale attention mechanism, boosting the model's ability to recognize targets of different sizes. However, these improvements also led to an increase in model parameters and size, though this did not affect deployment on UAV devices. Additionally, while the frame rate FPS of the improved model decreased, a frame rate above 120 frames per second is sufficient for UAV-based fire detection.

When C2f_DynamicConv and EMA were combined, the model not only improved in accuracy and mean average precision but also demonstrated a good balance in controlling parameter count and model size, with a precision of 75.2%, mAP50 of 68.1%, and mAP50-95 of 38.2%. The further introduction of Slimneck reduced the model size but had a slight impact on accuracy. Furthermore, the inclusion of Dynamic Head showed a positive effect on performance in some combinations, particularly in enhancing the model's detection capability for small targets.

Through ablation experiments, we found that when all modules were reasonably added to the model, it achieved optimal performance in terms of accuracy and mean average precision, with a precision of 75.7%, mAP50 of 69.5%, and mAP50-95 of 39.3%, while model size and parameter count remained within a reasonable range. This combination not only improved detection performance but also effectively controlled model size and parameter count, demonstrating a high cost-performance ratio. This finding has important guiding significance for model deployment and optimization in practical applications.

IV. Discussion

The improved YOLOv8n algorithm proposed in this study demonstrates significant performance enhancements in drone-based forest fire detection tasks. Its core innovation lies in achieving a synergistic improvement in detection accuracy and computational efficiency through a multi-dimensional optimization strategy. Compared to existing YOLO series models, this algorithm maintains a lightweight design while achieving an mAP50 of 69.5%, an improvement of 2.7% over the baseline model. This advancement holds substantial application value in the field of fire detection. The introduction of the dynamic convolution module effectively addresses the limitations of fixed convolution kernel parameters. By leveraging a dynamic parameter allocation mechanism, the network adapts to complex and dynamic fire scenarios, exhibiting superior feature extraction capabilities, particularly when dealing with challenges such as blurred smoke edges and irregular flame shapes.

TABLE III: Ablation Experiment of Attention Mechanism

Attention	P/%	mAP50/%	Params/M	FLOPs/G	FPS/$f \cdot s^{-1}$	Model Size/MB
Baseline	70.6	67.1	**3.0**	8.2	308.8	6.2
CBAM	72.4	67.3	3.8	**8.2**	491.9	6.1
ECA	71.9	67.3	**3.0**	**8.2**	255.2	**5.9**
CA	71.6	66.9	**3.0**	**8.2**	244.5	6.0
GAM	71.2	67.2	4.7	9.5	457.8	9.1
EMA	**72.5**	**67.5**	**3.0**	8.4	**498.9**	11.9

TABLE IV: Ablation Experiment

C2f_DynamicConv	EMA	Slim-neck	Dynamic Head	P/%	mAP50/%	mAP50-95/%	Params/M
				70.6	66.8	37.9	3.0
✓				73.3	67.6	37.7	4.4
	✓			74.7	67.6	37.6	3.0
		✓		71.5	67.7	38.4	2.8
			✓	73.5	67.9	38.6	4.7
✓	✓			75.2	68.1	38.2	4.4
	✓	✓		70.9	68.0	38.7	2.8
✓	✓	✓		75.0	68.0	38.9	3.6
✓	✓		✓	74.8	68.3	39.1	5.9
✓	✓	✓	✓	**75.7**	**69.5**	**39.3**	5.4

The application of the multi-scale attention mechanism significantly improves small-target detection performance. Its cross-spatial learning mechanism enhances the deep network's ability to retain subtle fire-related features through global information encoding. Experimental results show that the EMA module improves accuracy in small-target detection scenarios by 1.9%, making it crucial for early fire warning. Additionally, the introduction of the Slim-neck structure, utilizing depthwise separable convolutions and structural reparameterization techniques, optimizes feature fusion efficiency while increasing the parameter count by only 0.4M, validating the feasibility of lightweight design for deployment on drone platforms.

Compared to YOLOv9T, which has a similar parameter size, this method achieves a 0.7% improvement in mAP50-95, with significantly enhanced detection robustness in smoke-obscured scenarios. The Slim-neck module, through depthwise separable convolution and structural reparameterization techniques, reduces the neck network size to 10.6MB—9.4% smaller than YOLOv7t—offering feasibility for edge deployment.

Notably, while the model performs well on the public dataset C4-AI, there is still room for improvement under extreme lighting conditions, such as strong backlighting or nighttime infrared imagery. Additionally, the current model is primarily optimized for static flame detection, and its capability to track rapidly spreading forest fire lines has yet to be fully validated. Although there remains a risk of missed flame edge detection under extreme lighting conditions, integrating infrared spectral data fusion and adaptive exposure compensation mechanisms could further optimize performance. This research not only provides a cost-effective solution for real-time forest fire monitoring but also offers an innovative optimization paradigm for object detection tasks in resource-constrained environments.

V. Conclusions

This study addresses key challenges in drone-based forest fire detection by proposing a lightweight improved algorithm based on the YOLOv8n architecture. By optimizing feature extraction with the dynamic convolution module, enhancing multi-scale perception with DyHead, strengthening feature representation with the EMA attention mechanism, and implementing model compression through the Slim-neck structure, the model maintains a lightweight size of 10.6MB while improving mAP50 to 69.5% and achieving an accuracy of 75.7%. Experimental results show that the improved algorithm reduces the false detection rate by 37% under complex background interference and increases the recall rate for small-target detection by 21%, significantly outperforming existing mainstream models.

The core innovations of this technology include: (1) Establishing a dynamic parameter allocation mechanism to enable intelligent scheduling of computational resources; (2) Constructing a multi-dimensional attention system to enhance feature discrimination in complex scenarios; and (3) Innovating a lightweight architecture design to balance detection accuracy and computational efficiency. These advancements provide a reliable technical solution for real-time fire monitoring on drone platforms, particularly in complex terrain and resource-constrained forest fire prevention scenarios.

Future work will address three areas: (1) Developing a fusion mechanism to boost detection robustness in extreme environments; (2) Optimizing the dynamic inference framework for adaptive allocation of computational resources; and (3) Building a fire spread prediction model to expand system functionality. This study not only advances the application of object detection technology in forest fire prevention but also provides valuable methodological insights for edge computing tasks in other resource-constrained environments.

VI. Declarations

This work was supported in part by the National Natural Science Foundation of China under Grant 62171327, and in part by the 16th Postgraduate Innovation Fund of Wuhan Institute of Technology(CX2024146).

References

[1] K. H. Jodhani, H. Patel, U. Soni, R. Patel, B. Valodara, N. Gupta, A. Patel, and P. Omar, "Assessment of forest fire severity and land surface temperature using google earth engine: A case study of gujarat state, india," *Fire Ecology*, vol. 20, no. 1, p. 23, 2024.

[2] L. Chapungu, G. Nhamo, D. Chikodzi, and K. Dube, "Trends and impacts of temperature and fire regimes in south africa's coastal national parks: implications for tourism," *Natural Hazards*, vol. 120, no. 5, pp. 4485–4506, 2024.

[3] Y. Wang, Y. Wang, C. Xu, X. Wang, and Y. Zhang, "Computer vision-driven forest wildfire and smoke recognition via iot drone cameras," *Wireless Networks*, vol. 30, no. 9, pp. 7603–7616, 2024.

[4] B. Zhou, S. Zhang, R. Xue, J. Li, and S. Wang, "A review of space-air-ground integrated remote sensing techniques for atmospheric monitoring," *Journal of Environmental Sciences*, vol. 123, pp. 3–14, 2023.

[5] E. Anastasiou, A. T. Balafoutis, and S. Fountas, "Trends in remote sensing technologies in olive cultivation," *Smart Agricultural Technology*, vol. 3, p. 100103, 2023.

[6] Z. Fang and A. V. Savkin, "Strategies for optimized uav surveillance in various tasks and scenarios: A review," *Drones*, vol. 8, no. 5, p. 193, 2024.

[7] H. Ding, Q. Li, L. Zhao, L. Qi, S. Tang, M. Zhang, M. Li, and S. He, "Model and criteria for slope monitoring and early warning in open-pit mines using ground-based real aperture radar," *Environmental Earth Sciences*, vol. 83, no. 19, pp. 1–21, 2024.

[8] P. Liu and G. Zhang, "A case study on the integration of remote sensing for predicting complicated forest fire spread," *Remote Sensing*, vol. 16, no. 21, p. 3969, 2024.

[9] M. Li, H. Sun, and R. Zhao, "A review of root zone soil moisture estimation methods based on remote sensing," *Remote Sensing*, vol. 15, no. 22, p. 5361, 2023.

[10] A. Sabino, L. N. Lima, C. Brito, L. Feitosa, M. F. Caetano, P. S. Barreto, and F. A. Silva, "Forest fire monitoring system supported by unmanned aerial vehicles and edge computing: a performance evaluation using petri nets," *Cluster Computing*, vol. 27, no. 7, pp. 9735–9755, 2024.

[11] G. Li, P. Cheng, Y. Li, and Y. Huang, "Lightweight wildfire smoke monitoring algorithm based on unmanned aerial vehicle vision," *Signal, Image and Video Processing*, vol. 18, no. 10, pp. 7079–7091, 2024.

[12] K. Kılıç, K. Kılıç, İ. A. Doğru, and U. Özcan, "Wd detector: deep learning-based hybrid sensor design for wood defect detection," *European Journal of Wood and Wood Products*, vol. 83, no. 2, pp. 1–13, 2025.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[14] L. Zhang, M. Wang, Y. Ding, and X. Bu, "Ms-frcnn: A multi-scale faster rcnn model for small target forest fire detection," *Forests*, vol. 14, no. 3, p. 616, 2023.

[15] Y. Huang, X. Zhou, X. Xiong, and Y. Fu, "Efficient defect detection method for wire and arc additive manufacturing based on modified yolov8 model," *Journal of Nondestructive Evaluation*, vol. 44, no. 2, pp. 1–21, 2025.

[16] Y. Fei, F. Liu, M. Su, G. Wang, X. Sun, and R. Zang, "Real-time detection of small underwater organisms with a novel lightweight sfesi-yolov8n model," *Journal of Real-Time Image Processing*, vol. 22, no. 1, p. 23, 2025.

[17] B. R. Ilyas, B. M. Sofiane, T. A. Abderrazak, K. Miloud, F. Kamila, and B. Boukenadil, "Forest fire detection with combined svm and deep cnn approach," in *2024 2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC)*. IEEE, 2024, pp. 1–6.

[18] H. Wang, H. Qian, and S. Feng, "Ssd-kdgan: a lightweight ssd target detection method based on knowledge distillation and generative adversarial networks," *The Journal of Supercomputing*, vol. 80, no. 16, pp. 23 544–23 564, 2024.

[19] Z. Yang, Y. Shao, Y. Wei, and J. Li, "Precision-boosted forest fire target detection via enhanced yolov8 model," *Applied Sciences*, vol. 14, no. 6, p. 2413, 2024.

[20] Z. Xue, H. Lin, and F. Wang, "A small target forest fire detection model based on yolov5 improvement," *Forests*, vol. 13, no. 8, p. 1332, 2022.

[21] Z. Xiao, F. Wan, G. Lei, Y. Xiong, L. Xu, Z. Ye, W. Liu, W. Zhou, and C. Xu, "Fl-yolov7: A lightweight small object detection algorithm in forest fire detection," *Forests*, vol. 14, no. 9, p. 1812, 2023.

[22] S. Gao, G. Huang, X. Chen, H. Jiang, L. Zhou, and X. Gao, "Two-stage deep learning-based video image recognition of early fires in heritage buildings," *Engineering Applications of Artificial Intelligence*, vol. 129, p. 107598, 2024.

[23] H. Yar, Z. A. Khan, I. Rida, W. Ullah, M. J. Kim, and S. W. Baik, "An efficient deep learning architecture for effective fire detection in smart surveillance," *Image and Vision Computing*, vol. 145, p. 104989, 2024.

[24] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 030–11 039.

[25] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, and L. Zhang, "Dynamic head: Unifying object detection heads with attentions," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 7373–7382.

[26] D. Ouyang, S. He, G. Zhang, M. Luo, H. Guo, J. Zhan, and Z. Huang, "Efficient multi-scale attention module with cross-spatial learning," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[27] L. LIH *et al.*, "Slim-neckbygsconv: A betterdesign paradigm ofdetectorarchitecturesfor autonomous vehicles," *arXiv preprint*, 2022.

[28] S. Zhuohang, "Dynamic convolution-based image dehazing network," *Multimedia Tools and Applications*, vol. 83, no. 16, pp. 49 039–49 056, 2024.

[29] S. Chen, H. Sheen, T. Wang, and Z. Yang, "Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality," *arXiv preprint arXiv:2402.19442*, 2024.

[30] Y. Si, H. Xu, X. Zhu, W. Zhang, Y. Dong, Y. Chen, and H. Li, "Scsa: Exploring the synergistic effects between spatial and channel attention," *Neurocomputing*, vol. 634, p. 129866, 2025.