Smoke Detection Model with Adaptive Feature Alignment and Two-Channel Feature Refinement

Yuanpan Zheng*, Binbin Chen, Zeyuan Huang, Yu Zhang, Chao Wang, Xuhang Liu
School of Computer Science and Technology, Zhengzhou University of Light Industry, Zhengzhou, China

***Abstract*—To address issues of missed detections and low accuracy in existing smoke detection algorithms when dealing with variable smoke patterns in small-scale objects and complex environments, FAR-YOLO was proposed as an enhanced smoke detection model based on YOLOv8. The model adopted Fast-C2f structure to optimize and reduce the amount of parameters. Adaptive Feature Alignment Module (AFAM) was introduced to enhance semantic information retrieval for small targets by merging and aligning features across different layers during point sampling. Besides, FAR-YOLO designed an Attention Guided Head (AG-Head) in which feature guiding branch was built to integrate critical information of both localization and classification tasks. FAR-YOLO refines key features using Dual-Feature Refinement Attention module (DFRAM) to provide complementary guidance for the both two tasks mentioned above. Experimental results demonstrate that FAR-YOLO improves detection accuracy compared to existing. There's a 3.5% Precision increase and a 4.0% AP_{50} increase respectively in YOLOv8. Meanwhile, the model reduces number of parameters by 0.46M, achieving an FPS of 135, making it proper for real-time smoke detection in challenging conditions and ensuring reliable performance in various scenarios.**

***Keywords*—Smoke detection model; adaptive feature alignment; two-channel feature refinement; attention mechanism**

I. Introduction

Fires pose a major danger to human safety, economies and ecosystems. In 2023, there were 550,000 fire incidents reported in China within just six months, resulting in 959 deaths, 1,311 injuries, and property damage amounting to 3.94 billion yuan [1]. Between 2019 and 2020, Australia endured a forest fire that lasted more than seven months, killing billions of animals and destroying over 10 million hectares of land [2]. The best way to prevent the spread of fires is to suppress the spread of fires quickly and to disperse fire sources in a timely manner. However, early flames are small and can easily be obscured, so detecting the smoke generated by fires is the optimal approach for controlling the occurrence of fire.

Early smoke detection methods [3] relied on smoke, temperature, and light sensors to detect fire particles at close range, but had limited range and were prone to environmental interference. Traditional fire smoke detection algorithms relied on manual feature extraction and machine learning classification [4], but depended on domain expertise and couldn't effectively capture image features, resulting in poor generalization and applicability.

Object detection approaches based on deep learning can automatically learn main features and details in data, offering advantages such as high accuracy and strong robustness. Convolutional Neural Networks (CNNs) extract hierarchical features layer by layer through local connections and parameter sharing mechanisms, making them highly effective for image recognition. Recently, Transformer-based architectures have achieved remarkable advances in the field of image detection. Compared to CNNs, Transformers perform well in identifying distant relationships and perceiving global information within images. However, their computational complexity is higher, and they generally require more computing resources.

Xie et al [5], introduced a forest fire smoke identification method developed with the Faster-RCNN model, which enhances the receptive field by adding a feature fusion module after each level of the feature pyramid structure. However, this region extraction-based detection method consists of two stages, leading to higher algorithm complexity and slower detection speeds. YOLO series algorithms, on the other hand, are widely used for their capability to provide precise and timely detection. Casas et al [6], has shown that the excellent applicability of YOLO algorithms in smoke detection. Zhang et al [7] introduced an enhanced YOLOv4 model that combines an attention mechanism to boost the capture of smoke feature and utilizes the K-means++ algorithm to determine the most suitable predicted bounding box scale. Despite these improvements, the model suffers from slow detection speeds, which are inadequate for the real-time requirements for smoke detection. Li et al [8], added the YOLOv5 model with a coordinate attention mechanism to strengthen the model's concentration on key smoke regions and proposed an RFB module to capture global information. However, this model still struggles with detecting small smoke targets and exhibits a low smoke recognition rate. Ouyang et al [9], introduced a new object detection model named fuse-transformer, which combines Transformer and YOLOX to use transformer to handle global context and boost the model's potential to extract feature. However, the model has issues such as excessive size, high complexity, and demanding hardware requirements.

In the past decade, numerous algorithms have been developed for fire smoke detection, yielding promising results. However, challenges persist. First, the rapid spread of fires demands prompt smoke detection. Moreover, complex environmental conditions can alter the concentration and shape of smoke, making it harder for models to accurately identify it. Objects with colors and shapes similar to smoke may also lead to false detections. Additionally, the small size of early-stage smoke features poses another significant challenge. Prior studies used complex models to improve smoke detection accuracy. But large-parameter models are complex and slow.

To meet real-time demands, some applied one stage detection models with specific feature modules. However, single stage models struggle with small targets and complex environments. We aims to attain a desirable trade-off between speed and accuracy.

This paper presents an enhanced fire smoke detection model, which is built upon YOLOv8 [10], named FAR-YOLO (Feature Alignment and Refinement-YOLO). This model attains a balance between precision and speed by incorporating innovative lightweight modules and an attention mechanism-enhanced head structure. It utilizes an adaptive upsampling module to enhance capability of capturing small smoke targets. Additionally, we have constructed an outdoor fire smoke detection dataset consisting of 3,705 real smoke images. The dataset includes images of fire smoke captured at both close and distant ranges, as well as samples from complex environments with potential interference.

In this paper, we proceed as follows: Section II presents the relevant key technologies. Section III details the innovative methods, including the design philosophy and approach of the smoke feature extraction enhancement module. In Section IV, the datasets, experimental environment, ablation experiments and comparative experiments with other detection are models introduced. Section V summarizes the work content and contributions of this paper.

II. Related Work

A. YOLOv8

The network design of YOLOv8 is depicted in Fig. 1. The C2F module has cross connections between its layers and splitting operations, a design that enhances gradient fluidity and boosts the model backbone's efficiency in feature extraction. By introducing PANet [11] into the neck structure, the network can transfer features from bottom to top and from top to bottom, thereby effectively fusing multi-level semantic information and geometric information. YOLOv8 includes a decoupled head structure and incorporates Distribution Focal Loss [12] and IOU Loss in the localization branch, improving its ability to detect partially occluded objects. Additionally, a Task-Aligned Assigner [13] is used for sample matching, which evaluates both object localization and classification tasks, and then determines whether an instance is a target or irrelevant sample based on weighted scores, enhancing the model's performance across multiple tasks.

B. Attention Mechanism

The attention mechanism dynamically identifies key image regions and assigns positional weights. CBAM [14] enhances the representational power of the feature map by applying weighting to features across channel and spatial dimensions. Coordinate Attention (CA) [15] encodes the spatial coordinates to generate a coordinate weight map, which is then used to adjust the original feature map. Efficient Multi-Scale Attention (EMA) [16] addresses the accuracy loss that occur during dimensionality reduction in coordinate weight calculation by transmitting additional feature information across different regions.

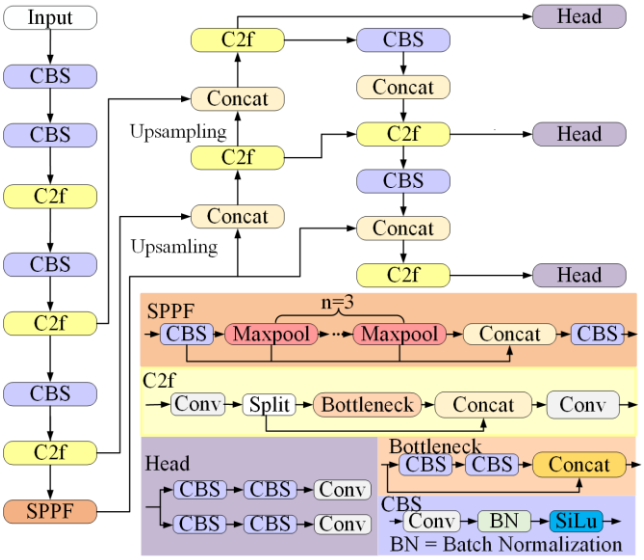


Fig. 1. Network architecture of YOLOv8.

C. Upsampling Methods

Upsampling methods boost image resolution and restore details. Bilinear interpolation is a widely used upsampling method that estimates the value of a target point using the positional information of neighboring points. CARAFE [17] dynamically generates adaptive kernels by perceiving the content of the features to reorganize input features. Dysample [18] uses a point sampling mechanism, dynamically calculating sampling point offsets to adapt to input feature maps. Compared to kernel-based methods, Dysample achieves better results and higher computational efficiency.

III. Improvement Scheme

The architecture of FAR-YOLO is depicted in Fig. 2. Fast-C2f is adopted instead of the original C2f structure, so that model complexity is reduced and detection speed is increased without losing accuracy. The lightweight AFAM module performs adaptive sampling during feature map reconstruction and is integrated into the upsampling process of the feature pyramid to enhance semantic information transfer across layers. In the head region, the proposed AG-Head detection head includes a feature guidance branch that consolidates key features from the two task branches. The DFRAM refines feature representations, guiding both classification and localization tasks.

A. Fast-C2f Module

In smoke detection, the model's inference speed is crucial. The calculation formula for *Latency* is as follows:

$$Latency = \frac{FLOPs}{FLOPS} \tag{1}$$

where, *FLOPs* is a measure of the total number of float computations, and *FLOPS* signifies the amount of these operations executed each second. Considering the high similarity between channels in the feature map, Chen et al [19], proposed Partial Convolution (PConv). As shown in Fig. 3(a),

PConv convolves only a segment of continuous channel images while keeping the other channels unchanged. Compared to regular convolution, PConv decreases parameters of the model and makes detection faster.

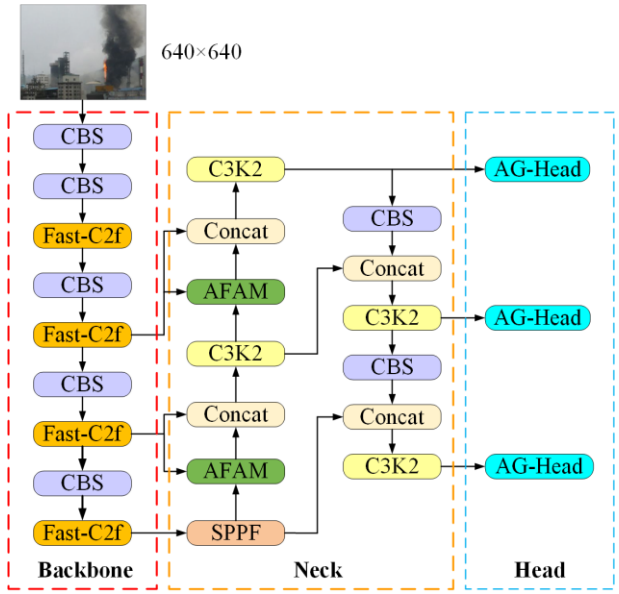


Fig. 2. Network architecture of FAR-YOLO.

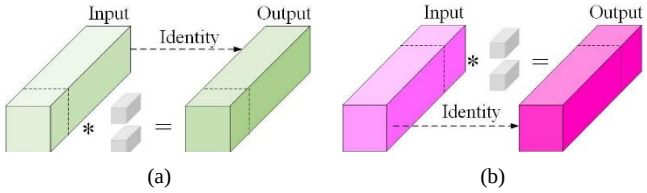


Fig. 3. Working way of Pconv: (a) Select the first quartile channel (b) Select the last quarter channel.

This paper proposes the Fast-C2f module, whose structure is presented in Fig. 4. The first partial convolution in the Fast-Bottleneck selects the first quarter of the channels for training. To avoid incomplete capture of key image information across all channels, we implemented an opposite channel selection scheme. Specifically, the second partial convolution selects the last quarter of the channels for training, as depicted in Fig. 3(b). Both of these complementary channel selection schemes enable Fast-Bottleneck to perform more comprehensive feature learning, providing the model with strong feature representation capabilities.

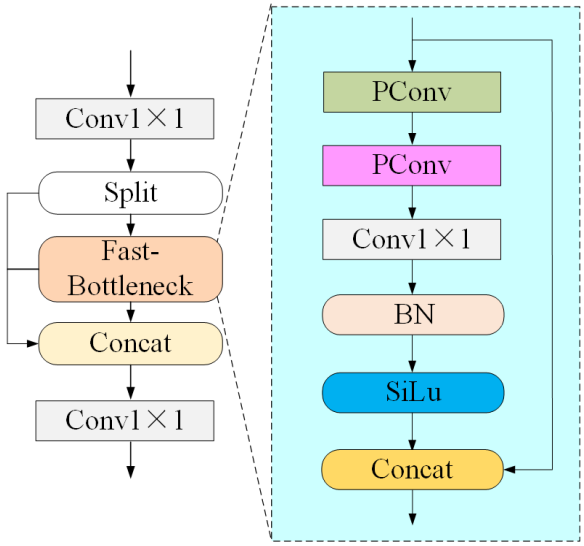


Fig. 4. Structure of Fast-C2f.

B. Adaptive Feature Alignment Module

Early-stage smoke has a small volume and covers merely a little pixel area in the image, resulting in limited appearance information. To address this, transferring detailed semantic information from deeper layers to shallower layers can enhance feature representation for small targets. This paper adopts this approach to improve the effectiveness of feature information transfer between deep and shallow layers. To avoid interference caused by feature mismatches during the upsampling process [20], we introduce the lightweight AFAM and apply it during the upsampling stage. This module's network framework is depicted in Fig. 5.

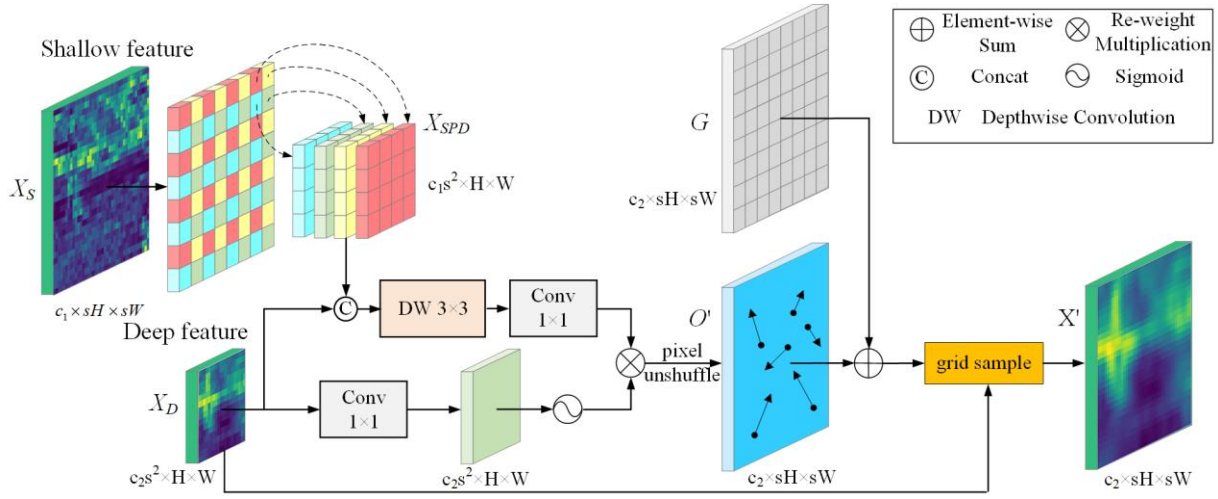


Fig. 5. Structure of AFAM.

AFAM employs point sampling for upsampling. Provided with an input feature map of size $c_2 s^2 \times H \times W$, the algorithm uses a linear layer to capture pixel neighborhood information, generating an offset flow O that reflects semantic variation trends between deep and shallow features. After reshaping O to $c_2 \times sH \times sW$, it combines with the sampling grid G to produce the sample set S. The following are the formulas for the calculation of O and S:

$$O = linear(X) \tag{2}$$

$$S = G + O \tag{3}$$

The $grid\_sample$ function is employed to resample the sample set, generating the final feature map X'.The formula for calculating X' is shown below:

$$\boldsymbol{X}' = grid\_sample(\boldsymbol{X}, \boldsymbol{S}) \tag{4}$$

AFAM incorporates shallow image features during the generation of O, improving feature alignment between adjacent layers by fusing features from different levels. The geometric details from the shallow layers help guide the deeper semantic information, thereby generating a more effective offset flow. In terms of implementation, the shallow image has dimensions of $c_1 s^2 \times sH \times sW$, and we aim to adjust its dimensions to match the deep feature map. Inspired by the SPD module [21], the specific approach is as follows: AFAM divides the shallow feature map into sub-maps of size $H \times W$, then reorganizes the objects at corresponding positions in each sub-map to form a feature map with the size of $c_1 s^2 \times H \times W$. The calculation formula is as follows:

$$\boldsymbol{X}_{SPD} = \begin{cases} f(0,0) = \boldsymbol{X}_S[0:H:s, 0:W:s] \\ f(1,0) = \boldsymbol{X}_S[1:H:s, 0:W:s] \\ f(s-1,0) = \boldsymbol{X}_S[s-1:H:s, 0:W:s] \\ f(0,s-1) = \boldsymbol{X}_S[0:H:s, s-1:W:s] \\ f(s-1,s-1) = \boldsymbol{X}_S[s-1:H:s, s-1:W:s] \end{cases} \tag{5}$$

where, X_S represents the shallow feature image, s is the scaling factor, and H and W are used to signify the horizontal and vertical extent of the feature map. This method effectively preserve the detailed information in the image.

To further enhance the effectiveness of feature fusion, AFAM applies linear projection and nonlinear transformation to the deep feature map to generate a weight map. The weight map is used to adaptively balance the feature information across different layers. The calculation formula is as follows:

$$O' = Sigmoid(linear_1(X_D)) \otimes linear_2(Ct(X_D, X_{SPD})) \tag{6}$$

where, X_D represents the deep feature image, $linear_1$ and $linear_2$ represent the linear projection operation on the deep feature map and the combined of Depthwise convolution (DWConv) and 1×1 convolution, respectively. Ct denotes the feature concatenation operation, $\otimes$ represents matrix multiplication.

C. Attention-Guided Head

1) The design of AG-HEAD: The decoupled detection head uses separate convolutional layers for localization and classification tasks. However, constrained by fixed kernel sizes, these layers only capture local features. In outdoor smoke detection scenarios, the texture features at the edges of smoke are often weak. Overemphasizing dense areas while neglecting sparse regions may misjudge the true smoke extent, reducing detection accuracy.

This paper designs AG-Head, whose network structure is shown in Fig. 6. The feature map extracts spatial feature information through DWConv, forming a feature guidance branch parallel to the other two branches. The localization and classification branches focus on learning different feature [22], while the feature-guided branch captures the features shared by both tasks during the back propagation process. The DFRAM is integrated into the feature guidance branch to fuse different types of features, providing complementary spatial information guidance for both branches. By enhancing the performance of both tasks, the model can comprehensively focus on smoke features, accurately capturing both smoke concentration and global information.

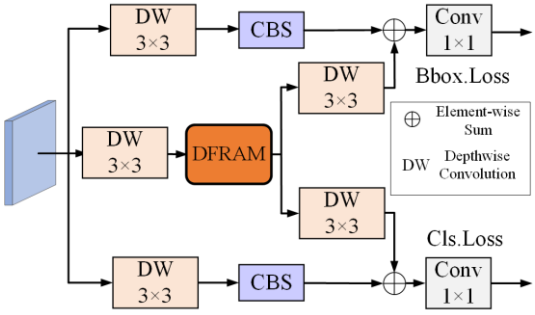


Fig. 6. Structure of AG-Head.

2) Dual-feature refinement attention module: To effectively fuse feature information and enhance the interaction between the two tasks, this paper proposes DFRAM, whose structure is shown in Fig. 7. This module contains two algorithms: Coordinate Feature Refinement (CFR) and Multi-Scale Feature Refinement (MSFR). CFR captures directional spatial location information, while MSFR extracts rich contextual and local features. DFRAM overlays the weight maps generated by both methods to enhance the detection head's sensitivity to smoke object concentration and spatial location.

a) Coordinate feature refinement: To get feature coordinate info, CFR first globally pools the input feature map vertically and horizontally. Then, CFR captures cross-channel interaction info in a special way. Since the fully connected method can't avoid the bad effects of channel dimensionality reduction and 1×1 2D convolutions aren't good enough for capturing inter-channel info, multi-kernel 1D convolutions are used to share channel info across n consecutive layers. The

calculation formulas for the global pooling operations in both directions are as follows:

$$GAP_H = \frac{1}{W} \sum_{o \le j < W} x(W, i) \tag{6}$$

$$GAP_W = \frac{1}{W} \sum_{o \le j < W} y(j, H) \tag{7}$$

The 1D convolution's kernel size is set based on the amount of channels, as the optimal kernel size is associated with the amount of channels [23]. The calculation formula is as follows:

$$n = |t|_{odd} = \left| \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right|_{odd} \tag{8}$$

where, C indicates the count of channels, and $|t|_{odd}$ represents the odd number adjacent to t.

b) Multi-scale feature refinement: MSFR extracts image features through DWConv and Dilated convolution (DConv) with progressively increasing dilation rates [24]. The architecture processes DWConv outputs in parallel through three DConv branches with diverse receptive fields, fuses these multi-scale features with the original input, and generates spatial attention weights via 1×1 convolution and Sigmoid activation. This design captures local details and contextual patterns for enhanced feature representation. The computation process of MSFR can be described as:

$$\boldsymbol{F} = DWConv(\boldsymbol{x_{input}}) \tag{9}$$

$$MSFR_{output} = Sigmoid(Conv_{1\times1}((\sum_{i=0}^{2} DConv_r(F)) + F)) \tag{10}$$

where, x_{input} is the original feature map and F denotes the intermediate layer's output, $MSFR_{output}$ refers to the output feature map of MSFR. $DConv_r$ represents dilated convolutions with varying rates in the three parallel branches, where subscript r indicates the dilation rate and i indicates the i branch in the multi-scale structure.

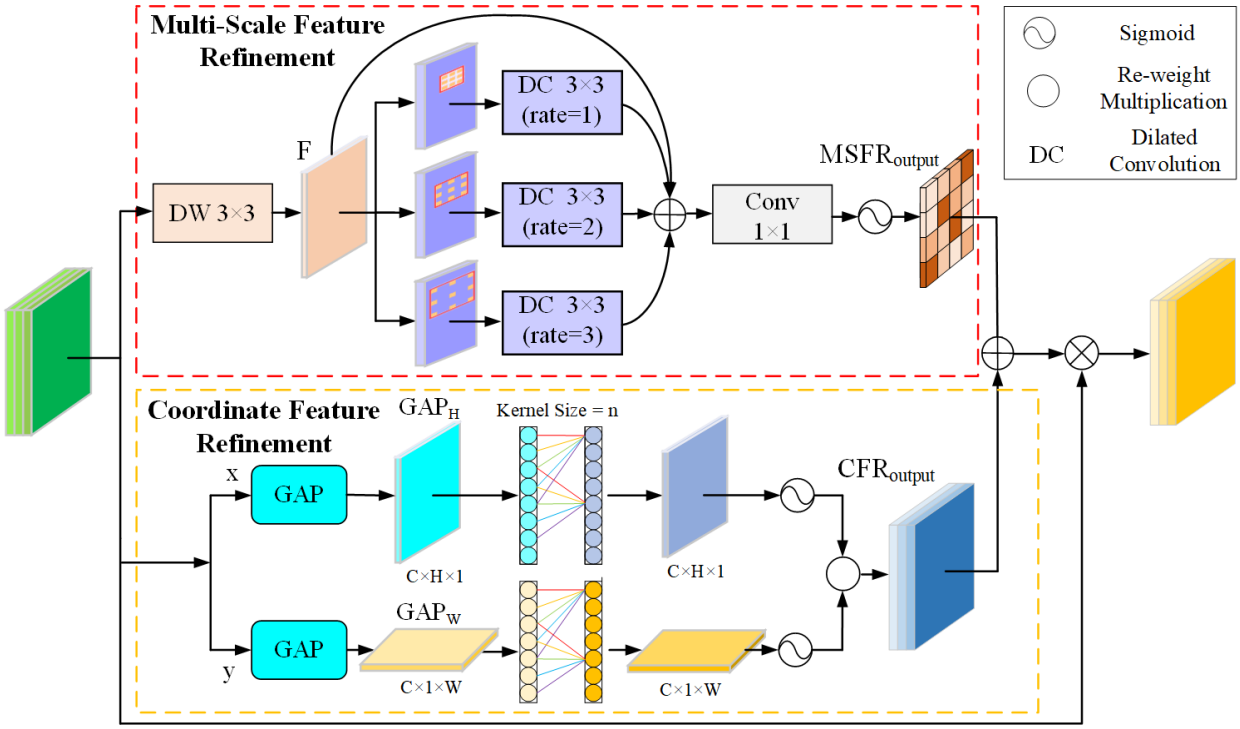


Fig. 7. Structure of DFRAM.

IV. Experiments and Result Analysis

A. Datasets and Annotations

Since there's a shortage of public fire smoke datasets and the lack of calibration, this paper collects smoke images and manually annotates the smoke regions to create a self-made fire smoke dataset. The smoke images are sourced from the public datasets HPWREN [25] and the Fire Detection Research Group [26], which contain real smoke objects of various sizes and in different scenes. This diverse dataset enables the model to develop strong recognition capabilities for various types of smoke. Images with low resolution or those not meeting the training requirements are excluded. The dataset is made up of 3,705 smoke images, which are randomly split into training, validation and testing sets in a 7:2:1 ratio. The dataset is formatted according to the YOLO dataset standard, and the LabelImg tool is used to annotate and create a plain text label file containing the positions and sizes of smoke targets. Some annotated image samples from the self-made dataset are shown in Fig. 8.

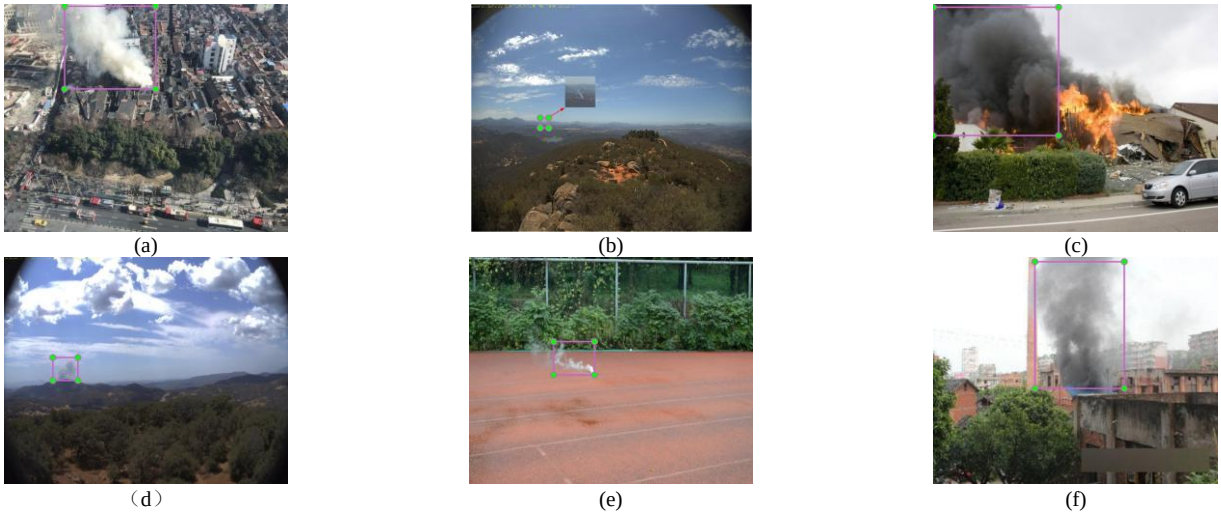


Fig. 8. Different types of labeled image samples from the self-made smoke detection dataset: (a) Large smoke; (b) and (f) show small smoke at different distances; (c) and (d) are black smoke; (e) is the smoke object in the environment of interference factors.

B. Evaluation Metrics

This paper uses COCO metrics to evaluate the smoke detection model. Precision reflects ratio of smoke samples that are correctly recognized, while recall is the proportion of actual smoke samples detected. Another important metric is Average Precision (AP), reflecting the model's overall accuracy in smoke identification. AP_{50} denotes the average precision at a threshold of 50. Additionally, Frames Per Second (FPS) measures indicates the rate at which a model can process consecutive images, and the number of parameters (Params) serves as a metric for assessing its complexity. The formulas for calculating these four metrics are presented below:

$$\mathrm{Pr}ecision = \frac{TP}{TP + FP} \tag{11}$$

$$\mathrm{Re}call = \frac{TP}{TP + FN} \tag{12}$$

$$AP = \frac{1}{r}\sum_{i=1}^{r} P_i \tag{13}$$

$$FPS = \frac{1}{Time} \tag{14}$$

where, TP indicates the count of smoke samples accurately detected, while FP signifies the amount of irrelevant samples mistakenly labeled as smoke objects, FN represents the amount of smoke samples that were missed, and $Time$ represents the time needed to process one image, which is measured in milliseconds.

C. Experimental Environment and Hyperparameter Configuration

Experiments use Python 3.8 and PyTorch 2.0, accelerated by CUDA 11.8. Hardware includes an AMD EPYC-7663X CPU and an NVIDIA GeForce RTX 3090 GPU. The optimizer is SGD. Input images are 640×640. The model is trained for 300 iterations, using batches of 16 and with a learning rate initially set to 0.001.

D. Comparative Experiment of Adaptive Feature Alignment Module

The upsampling operator can augment the model's proficiency in capturing essential information, but it also introduces complexity. Therefore, this section compares the AFAM module with the advanced lightweight CARAFE module and Nearest Neighbor Interpolation (NNI). The comparison results, presented in Fig. 9, demonstrates that the AFAM module achieves a Precision of 88.7% and an AP_{50} of 89.3% with fewer Params and FPS. Although CARAFE can achieve similar accuracy, it costs nearly twice as much in terms of Params and FPS compared to AFAM.

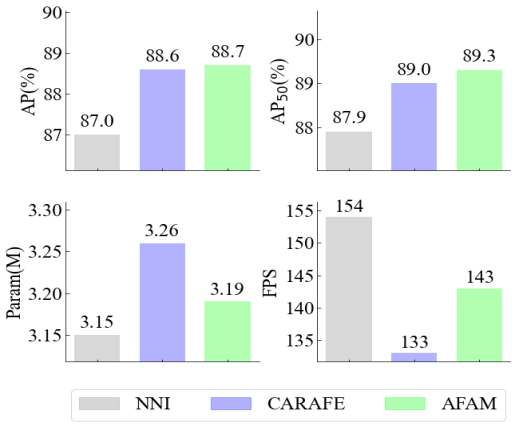


Fig. 9. Experimental comparison results of three upsampling methods under four different evaluation metrics: AP, AP_{50}, Params, and FPS.

E. Experimental Analysis of Dual-channel Feature Refinement Attention Module

This section investigates the influence of diverse dilation rate combinations of DConv in MSFR on the final results. Additionally, to evaluate the impact of the DFRAM, Several mainstream attention mechanisms are used for comparison with it.

1) Comparison experiments of different expansion rates: To study the impact of dilation rates on accuracy in MSFR, three combinations of dilation rates were tested: [1, 2, 3], [2, 4, 6], and [4, 8, 12]. These combinations were applied to MSFR, and experiments were conducted using the YOLOv8 network integrated with AG-Head on the self-made dataset. The comparison results are presented in Table I, illustrating that the [1, 2, 3] combination reaches the superior detection performance. While larger dilation rates capture a larger receptive field, they also result in losing local details, which reduces the model's capacity for recognizing small objects.

TABLE I. DETECTION RESULTS BASED ON DIFFERENT DILATION RATE COMBINATIONS' CONFIGURATIONS

Rates	*Precision (%)*	*AP_{50} (%)*
[1,2,3]	89.3	90.5
[2,4,5]	88.8	90.3
[4,8,12]	88.6	90.0

2) Comparative experiments with different attention modules: This section compares the performance of DFRAM with the CA, CBAM, and EMA. These attention modules are separately introduced into the YOLOv8 network integrated with AG-Head, and the attention maps generated by them are used to assist in task judgment. As shown in Table II, DFRAM shows the best performance.

TABLE II. DETECTION RESULTS BASED ON DIFFERENT ATTENTION MODULES' CONFIGURATIONS

Attention module	*Precision (%)*	*AP_{50} (%)*
CA	88.1	89.7
CBAM	88.9	90.1
EMA	89.2	90.3
DFRAM	89.3	90.5

F. Comparative Experiments

To assess the proposed improvements' effectiveness, this section conducts comparative experiments using the self-made smoke dataset, along with mainstream object detection methods from recent years. These methods include YOLO series detection algorithms such as YOLOv3 [27], YOLOv5 [28], and YOLOv7-tiny [29]. Additionally, the comparison includes the two-stage detection method Faster-RCNN [30] and cutting-edge algorithms based on the Transformer framework, such as Dino [31] and Dab-detr [32].

The experimental information is detailed in Table III. The YOLO series detection algorithms achieve higher accuracy than Faster-RCNN while requiring fewer parameters. Although the YOLO algorithms fall short of Transformer-based models in performance, their superior detection speed makes them more fitting for real-time smoke detection scenarios. Both YOLOv7-tiny and YOLOv8 achieved Precision exceeding 86.0%, with AP_{50} surpassing 87.6%. However, YOLOv7-tiny has parameters in an amount close to twice that of YOLOv8. The FAR-YOLO model not only achieves the best accuracy among the models tested but also requires fewer parameters than YOLOv8, demonstrating that FAR-YOLO offers the best overall performance.

TABLE III. COMPARATIVE EXPERIMENTS OF DIFFERENT ADVANCED MODELS

Compare Models	***Precision (%)***	***Recall (%)***	***AP_{50} (%)***	***Params (M)***	***FPS***
Faster-RCNN (ResNet50)	80.2	76.1	81.4	28.55	18
YOLOv3n	83.0	82.3	84.4	8.67	120
YOLOv5n	85.8	83.6	87.0	**1.89**	135
YOLOv7-tiny	86.1	84.0	87.6	6.20	123
YOLOv8n	87.0	84.3	87.9	3.15	**156**
Dino	89.7	87.3	91.2	47.00	17
Dab-detr	88.8	87.6	90.0	44.00	27
FAR-YOLO	**90.5**	**87.9**	**91.9**	2.69	135

A scatter plot intuitively compares model performance, with AP_{50} on the vertical axis and the number of Params on the horizontal axis. In the scatter plot, a model positioned further to the left indicates fewer parameters and lower computational complexity, while a position further up suggests higher precision. As illustrated in Fig. 10, Transformer-based models, despite their high precision, are located in the top right position due to their large number of parameters. This implies high computational resource demands, making them prone to deployment difficulties and slow operation on edge devices. In contrast, FAR-YOLO achieves a higher AP_{50} with fewer parameters, placing it in the top left region. Among models with similar parameter counts, FAR-YOLO is positioned higher, indicating superior performance at the same parameter level. Thus, FAR-YOLO strikes a good balance between precision and computational complexity, suits edge deployment, and can deliver ideal detection accuracy on resource-constrained edge devices with lower computational and storage costs.

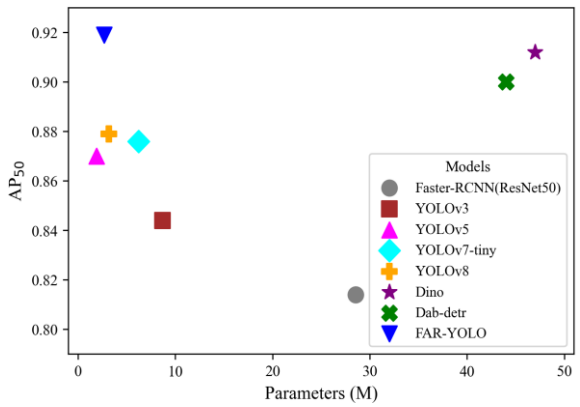


Fig. 10. Scatter plot of different models.

G. Ablation Experiments

This section presents ablation experiments on the self-made fire smoke dataset to assess the effect of the proposed improvements on model performance. The ablation experiment results are depicted in Table IV.

TABLE IV. ABLATION EXPERIMENTS FOR DIFFERENT MODULES

Experiment Group	Improvement Scenarios			Evaluation Metric			
	Fast−C2f	AFAM	AG−Head	*Precision (%)*	*Recall (%)*	*AP_{50} (%)*	*FPS*
1				87.0	84.3	87.9	156
2	✓			87.1	84.9	88.4	**161**
3		✓		88.7	86.1	89.3	143
4			✓	89.3	86.2	90.5	137
5		✓	✓	89.7	87.6	91.3	128
6	✓	✓	✓	**90.5**	**87.9**	**91.9**	135

In Group 2, the opposing channel allocation scheme of Fast-C2f increased the detection speed while maintaining the model's precision. The AFAM module enhanced the effectiveness of feature information transfer between different layers, resulting in a 1.4% increase in AP_{50} for Group 3. The introduction of the AG-Head, which includes the feature guidance branch and the DFRAM, resulted in Precision increasing by 2.3% and AP_{50} by 2.6% in Group 4. Group 5 combined AFAM and AG-Head, achieving a greater performance improvement compared to individual modules, with an AP_{50} of 91.3% and Recall of 87.6%, demonstrating that combining multiple modules yields better results. Finally, Group 6, which combined all three modules, showed a 4.0% increase in AP_{50} compared to Group 1, achieving a Recall of 87.9% and achieving a Precision of 90.5%. Although the FPS slightly decreased, it still reached 135 frames per second, reaching real-time detection requirements. Overall, the performance of the improved model demonstrated significant improvements.

The precision and recall curve evaluates precision and also takes recall into account across different thresholds, offering a comprehensive measure of model performance. Fig. 11 displays the precision and recall (pr) curves for the baseline model and various improvements. It is clear that the PR curve of the enhanced model largely overlaps with that of the baseline model, demonstrating that, at the same recall rate, the improved model achieves higher precision.

H. Detection Performance and Analysis

1) Visualization of improvement effects: To more effectively prove the validity of the proposed improvements, we use the YOLOv7tiny, YOLOv8 and FAR-YOLO models to detect smoke in fire scenes. As shown in Fig. 12, the improved model performs well in detecting large smoke plumes. The baseline model struggles to effectively recognize the entire smoke in scenarios involving large smoke with uneven concentration, often mistakenly dividing it into two parts. In contrast, the improved model captures richer contextual information, allowing it to accurately enclose the entire smoke plume with the detection box. Fig. 13 further demonstrates that the enhanced model surpasses the baseline in detecting small smoke targets. Additionally, under strong external light interference, the improved model can still accurately locate the smoke, while the baseline model fails to detect it, particularly in conditions of strong lighting and small smoke.

2) Visualization of smoke feature extraction capabilities: To better analyze the model's proficiency in smoke feature extraction, This paper adopts heatmaps to display the model's focus to different regions of the image during detection. The attention of a region is related to its color; the warmer the color, the higher the attention, indicating that the higher the attention, the greater the contribution of the region's features to the prediction result. As shown in Fig. 14, group (c) is the baseline model YOLOv8, and group (d) is the improved model FAR-YOLO. The improved model allocates more attention to the smoke region than the baseline model and the high-attention areas align with the contours of the complex-shaped smoke. This indicates that the improved model can accurately locate the region of interest, demonstrating superior performance.

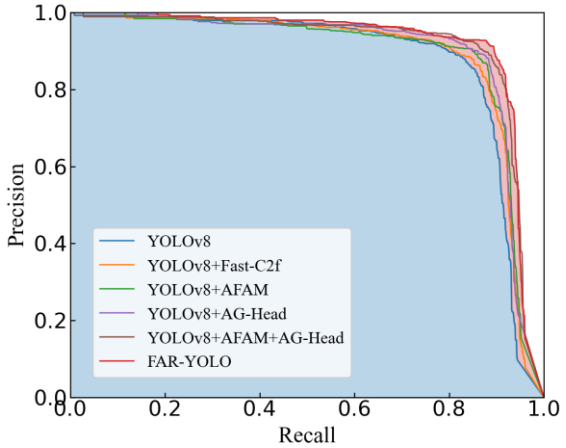
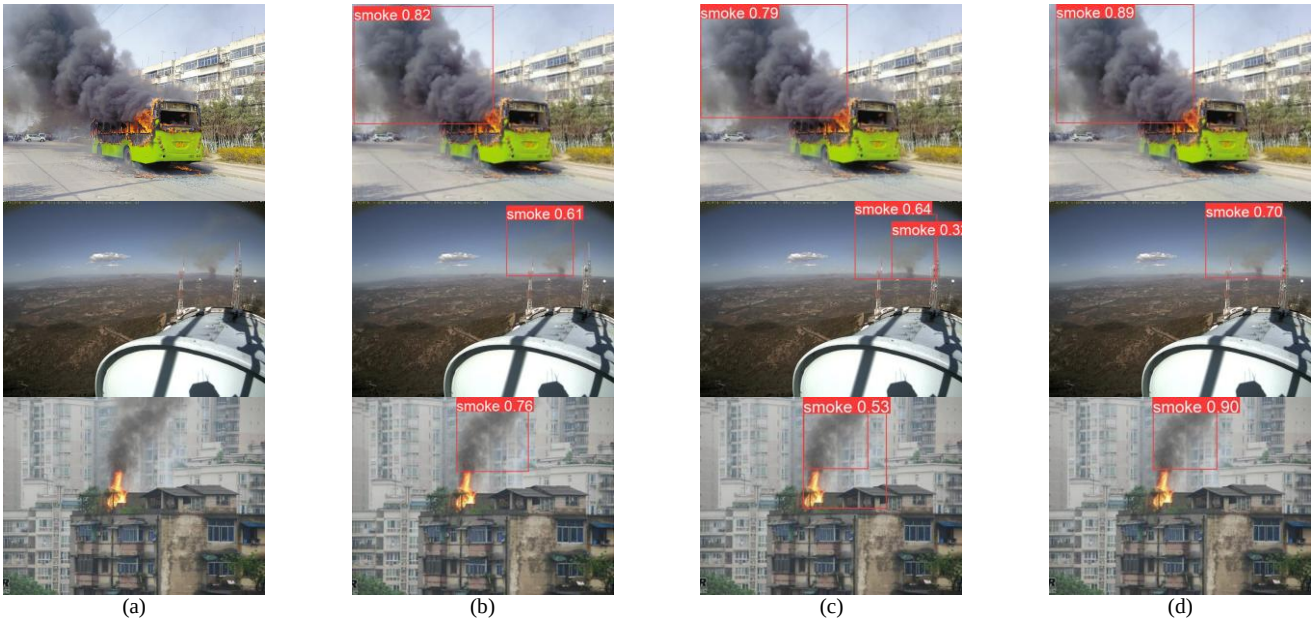
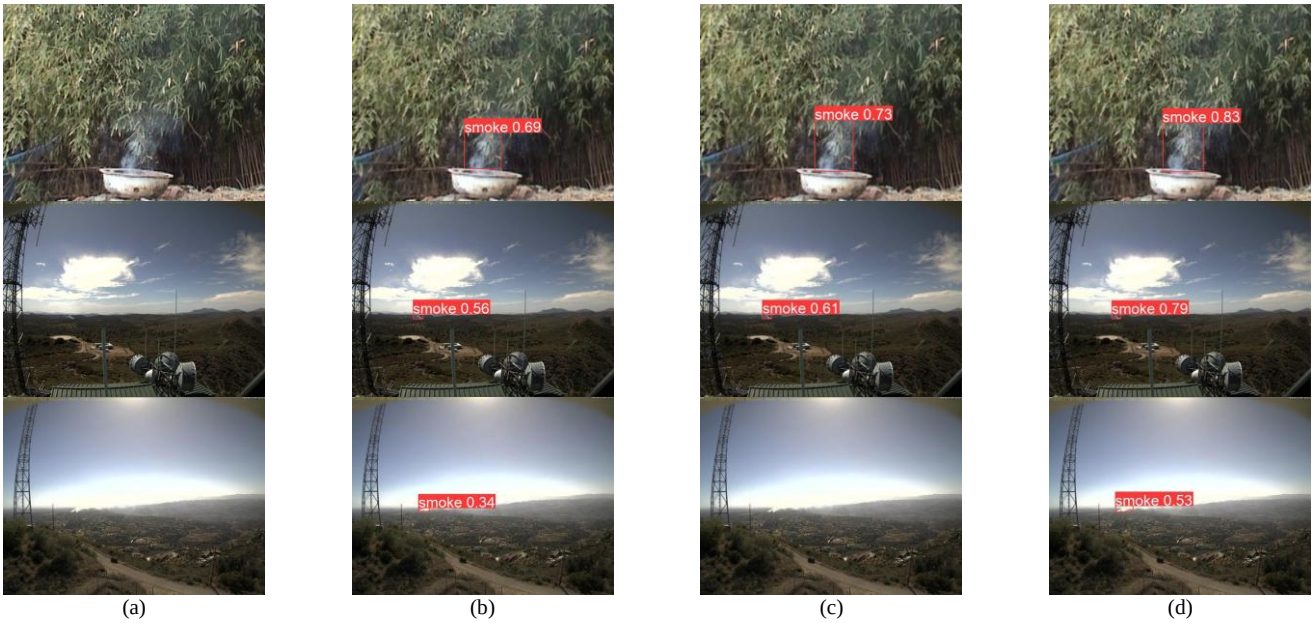


Fig. 11. Comparison of precision and recall (PR) curves with Different modules.



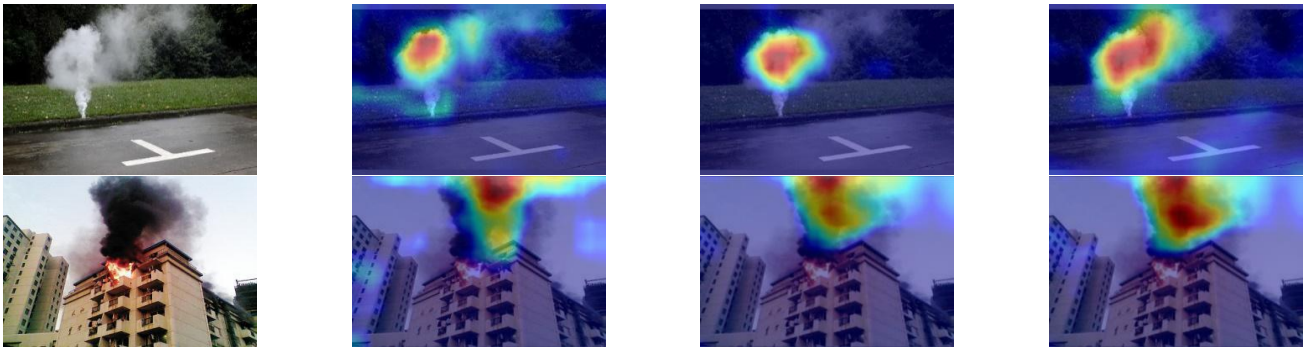
(a) (b) (c) (d)

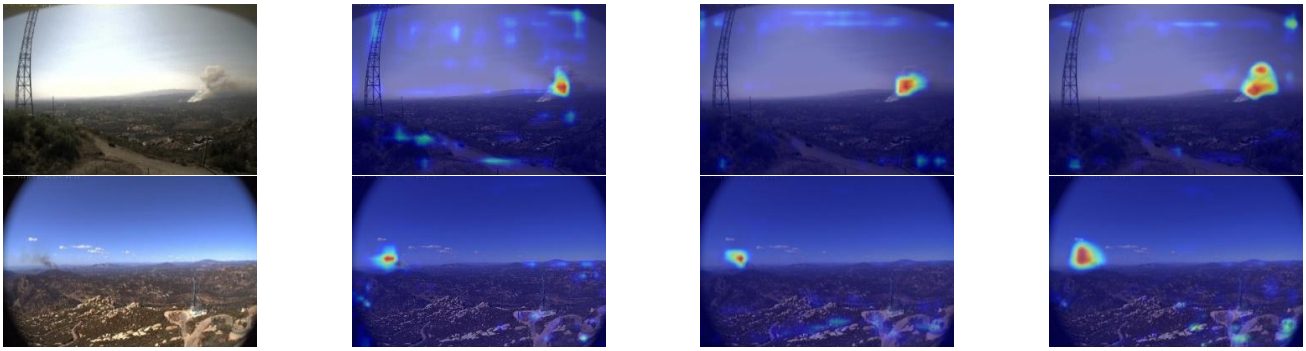
Fig. 12. Medium and large-scale smoke detection performance:(a) Original images;(b) YOLOv7tiny;(c) YOLOv8;(d) FAR-YOLO.



(a) (b) (c) (d)

Fig. 13. Early and small-scale smoke detection performance:(a) Original images;(b) YOLOv7tiny;(c) YOLOv8;(d) FAR-YOLO.





(a) (b) (c) (d)

Fig. 14. Heat maps showing the prediction of different models for smoke objects at near and far distances:(a) Original images; (b) YOLOv8; (c) FAR-YOLO.

V. CONCLUSION

This paper creates a multi-scene smoke dataset from public sources and introduces FAR-YOLO, an enhanced YOLOv8-based model. The model employs partial convolutions with two channel allocation strategies to build the Fast-C2f module, reducing complexity and boosting speed. The AFAM module is integrated into the upsampling process, uses adaptive alignment and resampling to strengthen the correlation between deep semantic and shallow positional features, improving small object detection. The AG-Head is introduced, featuring a feature-guided branch that extracts critical feature information from different task branches. The embedded DFRAM in this branch captures richer context and localization info, enhancing smoke concentration and scale judgment. Experiments show the model effectively detects multi-scale smoke in various scenes, with Precision and AP_{50} reaching 90.5% and 91.9%, respectively, and Recall achieving 87.9%. Additionally, the model reduces the parameter count by 0.46M and achieves a FPS rate of 135. The model effectively balances detection accuracy and speed, excelling in real-time smoke detection.

DATA AVAILABILITY

Data used for this article were collected by the research team and will be given to other researchers upon request.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

ACKNOWLEDGMENT

This study was supported by the Key Scientific Research Project Plan for Higher Education Institutions of Henan Province, China (No.25A520033).

REFERENCES

[1] "In the first half of 2023, the average daily number of fires across the country exceeded 3,000," National Fire and Rescue Administration, 2023. https://www.119.gov.cn/qmxfgk/sjtj/2023/38420.shtml

[2] Z. Xu,Y. Zhang, G. Blöschl, et al. "Mega forest fires intensify flood magnitudes in southeast Australia," Geophysical Research Letters, 2023, vol. 50, no. 12, pp. 1-10.

[3] X. Yang, L. Tang, H. Wang, et al. "Early detection of forest fire based on unmaned aerial vehicle platform," 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP) IEEE, 2019, pp. 1-4.

[4] A. Russo, K. Deb, S. Tista, et al. "Smoke detection method based on LBP and SVM from surveillance camera," 2018 International conference on computer, communication, chemical, material and electronic engineering (IC4ME2) IEEE, 2018, pp. 1-4.

[5] F. Xie, Z. Huang. "Aerial forest fire detection based on transfer learning and improved faster RCNN," 2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA) IEEE, 2023, vol. 3, pp. 1132-1136.

[6] E. Casas, L. Ramos, E. Bendek, et al. "Assessing the effectiveness of YOLO architectures for smoke and wildfire detection," IEEE Access, 2023, vol. 11, pp. 96554-96583.

[7] H. Zhang, Y. Hu, W. Ning. "Research on Smoke Detection Model Based on Improved YOLOv4," 2022 5th International Conference on Intelligent Autonomous Systems (ICoIAS) IEEE, 2022, pp. 1-6.

[8] J. Li, R. Xu, Y. Liu. "An improved forest fire and smoke detection model based on yolov5," Forests, 2023, vol. 14, no. 4, pp. 833.

[9] Z. Ouyang, Y. Wang, Z. Yin, et al. "Fusing Transformer and YOLOX for Smoke Detection," 2022 IEEE 22nd International Conference on Communication Technology (ICCT) IEEE, 2022, pp. 1740-1744.

[10] "YOLO by Ultralytic," Ultralytics, 2023. https://github.com/ultralytics/ultralytics

[11] S. Liu, L. Qi, H. Qin, et al. "Path aggregation network for instance segmentation," Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8759-8768.

[12] X. Li, W. Wang, L. Wu, et al. "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," Advances in neural information processing systems, 2020, vol. 33, pp. 21002-21012.

[13] C. Feng, Y. Zhong, Y. Gao, et al. "Tood: Task-aligned one-stage object detection," 2021 IEEE/CVF International Conference on Computer Vision (ICCV) IEEE Computer Society, 2021, pp. 3490-3499.

[14] S. Woo, J. Park, J. Lee, et al. "Cbam: Convolutional block attention module," Proceedings of the European conference on computer vision (ECCV), 2018, pp. 3-19.

[15] Q. Hou, D. Zhou, J. Feng. "Coordinate attention for efficient mobile network design," Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 13713-13722.

[16] D. Ouyang, S. He, G. Zhang, et al. "Efficient multi-scale attention module with cross-spatial learning," ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) IEEE, 2023, pp. 1-5.

[17] J. Wang, K. Chen, R. Xu, et al. "Carafe: Content-aware reassembly of features," Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 3007-3016.

[18] W. Liu, H. Lu, H. Fu, et al. "Learning to upsample by learning to sample," Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 6027-6037.

[19] J. Chen, S. Kao, H. He, et al. "Run, don't walk: chasing higher FLOPS for faster neural networks," Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 12021-12031.

[20] J. Wu, Z. Pan, B. Lei, et al. "FSANet: Feature-and-Spatial-Aligned Network for Tiny Object Detection in Remote Sensing Images," IEEE

Transactions on Geoscience and Remote Sensing, 2022, vol. 60, pp. 1-17.

[21] R. Sunkara, T. Luo. "No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects," Joint European conference on machine learning and knowledge discovery in databases Springer, 2022, pp. 443-459.

[22] Y. Yang, M. Li, B. Meng, et al. "Rethinking the misalignment problem in dense object detection," Joint European Conference on Machine Learning and Knowledge Discovery in Databases Springer, 2022, pp. 427-442.

[23] Q. Wang, B. Wu, P. Zhu, et al. "ECA-Net: Efficient channel attention for deep convolutional neural networks," Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11534-11542.

[24] F. Yu, V. Koltun. "Multi-scale context aggregation by dilated convolutions," arXiv preprint arXiv:1511.07122, 2015.

[25] HPWREN. "The HPWREN Fire Ignition images Library for neural network training," 2022. https://hpwren.ucsd.edu/FIgLib/

[26] Z. Qixing. "Research Webpage about Smoke Detection for Fire Alarm: Datasets," 2017. http://smoke.ustc.edu.cn/index.htm

[27] J. Redmon, A. "Farhadi. Yolov3: An incremental improvement," arXiv preprint arXiv:180402767, 2018.

[28] "Yolov5," Ultralytics, 2021. https://github.com/ultralytics/yolov5

[29] C. Wang, A. Bochkovskiy, Liao H. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 7464-7475.

[30] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2017, vol. 39, no. 6, pp. 1137-1149.

[31] H. Zhang, F. Li, S. Liu, et al. "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," arXiv preprint arXiv:220303605, 2022.

[32] S. Liu, F. Li, H. Zhang, et al. "Dab-detr: Dynamic anchor boxes are better queries for detr," arXiv preprint arXiv:220112329, 2022.