World Journal of Engineering Research
ISSN: 2959-9873
DOI: https://doi.org/10.61784/wjer3031

IMPROVING SMALL FIRE TARGET DETECTION IN UAV IMAGERY: AN ENHANCED RT-DETR WITH MULTI-SCALE FUSION AND EXPERT ROUTING

ZhiCheng Zhang
Queen Mary School Hainan, Beijing University of Posts and Telecommunications, Beijing 100876, China.
Corresponding Email: zzc040214@gmail.com

Abstract: Early fire detection is of paramount importance for forest fire prevention, yet traditional monitoring methods (e.g., satellites and ground-based stations) suffer from poor real-time performance or limited coverage. Unmanned aerial vehicles equipped with computer vision offer a novel solution for fire detection, but complex backgrounds, small flame and smoke targets, and varying illumination and weather conditions make accurate recognition challenging. In this work, we enhance the real-time detection Transformer model RT-DETR by designing a hybrid encoder architecture tailored for UAV fire imagery. Key improvements include the integration of an Adaptive Spatial Feature Fusion (ASFF) module to reconcile multi-scale feature inconsistencies; incorporation of Efficient Channel Attention (ECA) to strengthen channel-wise representations; replacement of the Transformer's fully connected feed-forward network with a Gated Mixture-of-Experts (MoE) structure to boost model capacity; and a multi-layer Transformer feature aggregation strategy. We evaluate the improved model on a UAV smoke fire dataset. Results show a significant uplift in both detection accuracy and recall: at an IoU threshold of 0.5, the enhanced RT-DETR achieves over 88.8% mAP—an approximate 2% gain over the original RT-DETR and superior performance compared to YOLO-series baselines. Ablation studies confirm that ASFF fusion, multi-attention mechanisms, and the MoE architecture each contribute meaningfully to small-target fire detection. Crucially, these advances incur negligible additional inference latency, enabling real-time intelligent monitoring for wildland fire scenarios.
Keywords: Fire detection; Real-time object detection; RT-DETR; Adaptive Spatial Feature Fusion (ASFF); Mixture-of-experts (MoE)

1 INTRODUCTION

Forest and wildland fires are severe natural disasters that not only threaten ecological environments and human life and property, but also exacerbate global warming through carbon emissions. Timely and accurate fire detection is crucial for disaster prevention and mitigation. However, traditional fire monitoring primarily relies on ground lookout towers, satellite thermal imaging, and other methods, which suffer from limited monitoring coverage or poor timeliness. For example, while satellite remote sensing can monitor large areas, it cannot provide early warnings during the initial stages of fires due to imaging cycle limitations[1]; ground monitoring stations and manual patrols are constrained by terrain and incur high costs. In recent years, with the development of unmanned aerial vehicle (UAV) technology, using UAVs equipped with visible light/infrared cameras for high-altitude patrols has provided new solutions for early fire detection. UAVs can fly flexibly at low altitudes, capturing fire scene images from multiple angles and enabling high-frequency patrol monitoring of forest areas. However, since fire targets (open flames or smoke) in UAV aerial images are often small in scale, irregular in shape, and easily confused with backgrounds, this poses significant challenges for automatic image-based detection. Complex forest backgrounds, occlusion, lighting changes, and the similarity between smoke and fog can all lead to missed detections and false alarms[2]. Therefore, research on detection algorithms specifically designed for UAV fire images is of great significance.

In recent years, deep learning has achieved breakthrough progress in computer vision object detection. Single-stage detectors (such as the YOLO series[3][4][5]) and two-stage detectors (such as Faster R-CNN[6]) have shown excellent performance in general object detection. However, directly applying these models to fire detection still faces difficulties: on one hand, fire datasets are relatively small and diverse in scenarios, prone to overfitting or unstable detection; on the other hand, existing detection models have insufficient capability for detecting small-scale targets and indistinct features, and direct application tends to produce high false negative rates. To improve wildfire recognition effectiveness, many scholars have made targeted improvements to existing detection architectures. For example, Mukhiddinov et al.[6] proposed an optimized early smoke detection model based on YOLOv5, improving average precision on their custom dataset to 73.6% through strategies such as improved anchor clustering, introducing SPP-Fast modules, and bidirectional feature pyramids. Yue Geng et al. integrated deformable convolution and BiFormer attention modules into YOLOv8 to enhance the extraction of flame and smoke features at different scales and suppress background interference, while adding a dedicated small target detection layer, resulting in a 1.3% improvement in model mAP_{50}, 1.5% improvement in precision, and 0.4% improvement in recall. These works demonstrate that incorporating multi-scale feature fusion and attention mechanisms into existing detection frameworks can effectively improve fire and smoke detection capabilities.

Concurrently, Transformer-based architectures have begun to make inroads into object detection. DETR, the pioneering approach by Carion et al.[7], formulates detection as a direct set-prediction problem using a Transformer encoder–decoder, obviating non-maximum suppression but suffering from slow convergence and suboptimal small-object performance. Subsequent efforts have augmented DETR with feature pyramids for multi-scale awareness[8], anchor-based queries, and improved query initialization[9]. In 2023, Baidu Research introduced Real-Time Detection Transformer (RT-DETR)[10], the first end-to-end Transformer detector capable of real-time inference. By combining a convolutional backbone with an efficient hybrid Transformer encoder—designed to decouple intra-scale modeling from cross-scale interactions—RT-DETR dramatically reduces computational overhead, achieving YOLO-comparable inference speeds. With IoU-aware query initialization, it attains 53.1 % mAP on COCO (with a ResNet-50 backbone) at 108 FPS, proving that Transformer detectors can meet real-time, small-object detection demands.

Despite these advances, RT-DETR still exhibits limitations in complex, small-target scenarios. Its simple layer-wise feature interactions may underutilize complementary information across scales; it lacks explicit channel-wise attention, leaving redundant background features unfiltered; and its shallow Transformer encoder, optimized for speed, constrains representational capacity needed to capture diverse fire patterns. To overcome these challenges, we propose an improved RT-DETR architecture for UAV-based fire detection. Our approach enriches the hybrid encoder with an adaptive multi-scale feature fusion module and an efficient channel-attention mechanism to strengthen representation of heterogeneous fire targets, and replaces the standard feed-forward network with a gated Mixture-of-Experts structure that increases model capacity while activating only a subset of experts to preserve real-time performance.

We validate our model on a proprietary UAV smoke fire dataset, comparing against the original RT-DETR and other leading detectors. Results demonstrate superior precision and recall, and ablation studies isolate the contributions of each enhancement. We also analyze the impact of our modules on parameter count and inference speed. The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 details the proposed model architecture; Section 4 describes experimental setup and results; Section 5 discusses the implications of our findings; and Section 6 concludes and outlines future research directions.

2 RELATED WORK

2.1 Fire Detection Methods

Early fire detection relied on traditional image processing and machine learning methods, such as utilizing color thresholds, motion detection, and background subtraction to identify flame or smoke regions[11]. However, these methods exhibited poor robustness to environmental variations, with high rates of false positives and false negatives. With the rise of deep learning, Convolutional Neural Network (CNN)-based approaches have become mainstream. Chen et al[12]. utilized convolutional neural networks to extract forest fire smoke features, achieving faster and more accurate recognition compared to traditional methods. Li Jie et al. and Feng Lujia et al[13]. further applied CNNs to flame and smoke detection tasks, proposing fire recognition algorithms and object region-based smoke recognition methods respectively, achieving high accuracy in laboratory environments. However, these methods mostly target static image classification or simple scenarios, and their performance remains unsatisfactory for small object detection in complex outdoor scenes.

Currently, the most effective fire detection methods are predominantly based on improvements to mainstream object detection frameworks. One category consists of two-stage detectors, with Faster R-CNN[14] as a typical representative. It first generates candidate boxes using a Region Proposal Network (RPN), then performs classification and refinement, with convolutional feature extraction at each stage, resulting in high detection accuracy but slower speed. In fire detection, some studies have applied Faster R-CNN to smoke detection with certain effectiveness, but the problem of small object missed detection persists. Another category comprises single-stage detectors, such as RetinaNet and the YOLO series. These methods directly regress detection boxes and classifications on densely sampled feature maps, offering faster speeds. The YOLO series has evolved rapidly, from YOLOv3 to YOLOv5, YOLOv7, and YOLOv8, continuously improving accuracy and speed. However, CNN-based architectures like YOLO still have limitations when dealing with large-scale variations and complex backgrounds, with their feature fusion and long-range dependency modeling capabilities being inferior to Transformer architectures.

2.2 RT-DETR and Transformer Detectors

Transformer initially achieved success in natural language processing, and Carion et al. introduced it to computer vision, proposing the first end-to-end object detection Transformer model, DETR[7]. DETR performs global modeling on CNN-extracted features through a Transformer encoder-decoder, directly outputting a set of bounding boxes and categories without requiring NMS post-processing. Despite its conceptual simplicity, the original DETR suffers from several issues: the model requires extremely long training time to converge, primarily due to the use of fixed random queries that make learning difficult; additionally, it performs poorly on small objects because Transformer processing of high-resolution features is computationally expensive.

The emergence of RT-DETR [10]addresses the bottleneck of Transformer detectors in real-time applications. Its core is an efficient hybrid encoder architecture: first employing a CNN backbone to extract multi-scale features (pyramid levels such as C3, C4, C5), then efficiently fusing these features through a hybrid encoder module. Unlike DETR's direct global self-attention on long sequences of flattened multi-scale features, RT-DETR decouples intra-scale feature

modeling from cross-scale feature fusion, significantly reducing encoder computational overhead. Specifically, the RT-DETR encoder first models local relationships using self-attention within each scale, then fuses information across different scales through lightweight modules. This design is termed "AIFI+CCFM" (Adaptive Intra-scale Feature Interaction + Cross-scale Feature Fusion Module). Meanwhile, RT-DETR introduces an IoU-aware query selection mechanism in the decoding stage, selecting features with high localization confidence from encoded features as initial queries, thereby improving detection accuracy. Thanks to these innovations, RT-DETR achieves accuracy comparable to or better than real-time detectors like YOLOv7-L while maintaining 108 FPS inference speed. It can be anticipated that Transformer architectures have broad application prospects in specific object detection tasks such as fire detection.

2.3 Mixture-of-Experts (MoE) Mechanism

Mixture-of-Experts is a machine learning concept from the 1990s that has recently resurged in large-scale neural networks. Instead of using one massive model for all inputs, MoE trains multiple "expert" sub-models with a gating network dynamically selecting a subset of experts based on input features. This allows large total parameters while activating only a few experts per inference, achieving enhanced model capacity with manageable computational overhead. Shazeer et al[15]. introduced sparse gating in Google's translation model, enabling billion-parameter training. Fedus et al.[16] proposed Switch Transformer, simplifying MoE routing by activating single experts, significantly reducing communication costs and improving stability. MoE has shown success in NLP through "conditional computation" and is gaining attention in computer vision. For example, Riquelme et al. proposed V-MoE[17] for Vision Transformers, achieving improved accuracy with reduced computation in image classification. Recent work by Yuan et al.[18] has also explored similar efficiency principles in ensemble learning, proposing a margin-maximizing fine-grained ensemble method that achieves superior performance with significantly fewer base learners through learnable confidence matrices and category-specific optimization. A key challenge is routing imbalance, typically addressed through load balancing losses. For fire detection, where flame and smoke appearance varies significantly across scenarios, MoE mechanisms could enable specialized experts for different fire feature types, improving overall detection performance.

2.4 Adaptive Multi-scale Fusion and Attention Mechanisms

Multi-scale feature fusion is crucial in object detection. While FPN structures fuse high and low-level features through top-down pathways, they typically use fixed weighting. ASFF (Adaptively Spatial Feature Fusion) learns position-wise fusion weights for different scale features, selecting the most informative scale at each pixel. Liu et al. proposed ASFF to address feature conflicts between layers in single-stage detectors, improving multi-scale prediction reliability through learned spatial filtering. ASFF significantly improves small object AP in models like YOLOv3 with minimal inference overhead. This study incorporates ASFF concepts in RT-DETR's feature fusion through lightweight spatial weight modules, enabling optimal high-low level feature combination for fire and smoke detection.

For attention mechanisms, SE channel attention and CBAM have proven effective in vision tasks. Considering the need to distinguish subtle differences between flames and smoke, we incorporate ECA (Efficient Channel Attention) modules in backbone feature extraction. ECA achieves efficient channel weight allocation through 1D convolution after global pooling without additional fully connected layers like SE. ECA enhances attention to useful feature channels with minimal parameter overhead and brings significant performance gains with negligible complexity increase. In fire detection, ECA helps highlight flame/smoke feature responses while suppressing background noise. Additionally, we adopt dynamic sparse attention from BiFormer, computing attention efficiently only for key queries in the Transformer encoder, reducing interference from irrelevant background tokens.

In summary, related research indicates that addressing UAV fire detection challenges requires integrating multi-scale features, focusing on effective information, and improving model expressiveness and robustness. Based on these insights, the next section introduces how our improved RT-DETR model organically combines ASFF, ECA, MoE, and other modules to enhance fire object detection performance.

3 PROPOSED METHODS

The overall architecture of the improved RT-DETR fire detection model proposed in this study is shown in Figure 1. The model is based on the RT-DETR framework and consists of three main components: a convolutional backbone network, a hybrid Transformer encoder, and a detection decoder. Our innovations are concentrated in the design of the hybrid encoder structure, including:

1) multi-scale feature adaptive fusion modules ASFF-2 and ASFF-3;
2) CSPRep residual layers fused with ECA attention;
3) gated mixture-of-experts routing Transformer encoder layers;
4) integration of multi-level Transformer features.

These modules will be described in detail below.

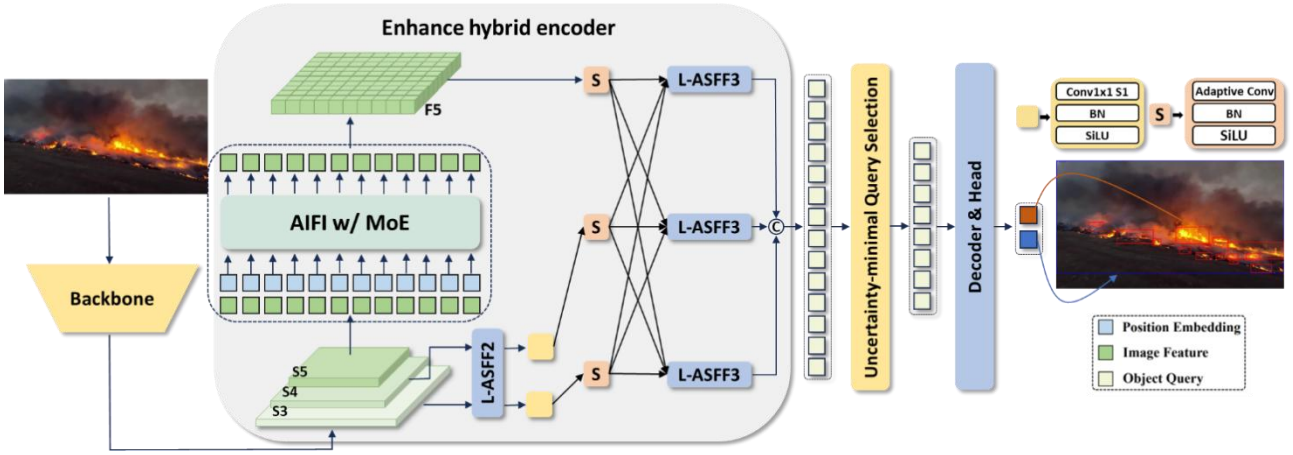

Figure 1 Schematic Diagram of the Improved RT-DETR Fire Detection Model Architecture. The Hybrid Encoder Contains Multi-Scale Fusion Modules Lightweight ASFF and MoE attention Transformer Layers

3.1 Adaptive Multi-scale Fusion and Attention Mechanisms

We employ ResNet18 convolutional network as the backbone for extracting multi-scale feature pyramids from images. ResNet18 contains 5 stages with output feature strides of 2, 4, 8, 16, and 32 respectively. We select the feature maps from the last three stages C_3, C_4, C_5 (with approximately 256, 512, 1024 channels respectively) for subsequent encoder use, which is consistent with the original RT-DETR configuration. Considering that the Transformer encoder expects unified dimensional input, we first compress the channels of each layer feature to a unified hidden space dimension D (such as 256) through 1×1 convolution, formulated as: $P_i = \mathrm{BN}(\mathrm{Conv}_{1\times1}(C_i))$, where P_i is the compressed i-th layer feature, and BN is the batch normalization layer. The obtained P_3, P_4, P_5 correspond to feature maps with high, medium, and low spatial resolutions respectively, representing different scale information of the image. Additionally, we introduce Efficient Channel Attention (ECA) in the residual blocks of each stage of the backbone network. The specific approach is: for the feature X output by the residual block, we first perform global average pooling to obtain channel description $z \in \mathbb{R}^C$, then apply one-dimensional convolution $\mathrm{Conv1d}(k)$ (where k is the kernel size, such as 3) for local interaction in the channel dimension, and finally use Sigmoid activation to obtain channel weights $\boldsymbol{\alpha} \in (0,1)^C$. We apply $\boldsymbol{\alpha}$ back to the original feature: $X' = \boldsymbol{\alpha} \odot X$ (element-wise multiplication by channel). The ECA module efficiently models inter-channel correlations and enhances the response of salient features of fire targets. We integrate ECA into the CSPRepLayer module implementation, which will be described in detail in Section 3.2.

3.2 CSPRep Residual Blocks and RepVGG Structure

After backbone feature compression, we design improved residual blocks for further feature refinement and coordination with ASFF fusion. We adopt the grouped residual structure concept from CSPNet, splitting the input features into two paths: one part goes through several stacked RepVGG Blocks to extract local new features, while the other part is retained as a shortcut, then they are fused by addition in the channel dimension. The RepVGG Block is the basic unit of the RepVGG network, consisting of a 3×3 convolution and a 1×1 convolution connected in parallel, with their outputs added together and passed through an activation function. During training, two branches are maintained, while during inference, the convolution kernels can be fused equivalently into a single convolution for inference acceleration. The CSPRepLayer module is formulated as:

- $X_1 = \mathrm{Conv}1\times1^{in\to h}(X)$, $X_2 = \mathrm{Conv}1\times1^{in\to h}(X)$ are the two branches that compress the input X to h channels respectively;
- $Y_1 = \mathrm{RepVGGBlock}_1(\mathrm{RepVGGBlock}_2(...(X_1)...))$ is the output of stacking N RepVGG residual blocks on X_1;
- Add the other branch X_2 with Y_1: $Z = Y_1 + X_2$;
- Apply channel attention to Z: $Z' = \mathrm{ECA}(Z)$;
- If the output channels need to be expanded to out, then transform through $\mathrm{Conv}_{1\times1}^{h\to out}$.

CSPRepLayer achieves the refinement of new features through multiple RepVGG blocks while retaining part of the original features, and adjusts channel weights using ECA. It enhances feature expression while controlling computational complexity. In our hybrid encoder, features after ASFF fusion pass through a CSPRepLayer to integrate information and prepare for the next stage processing.

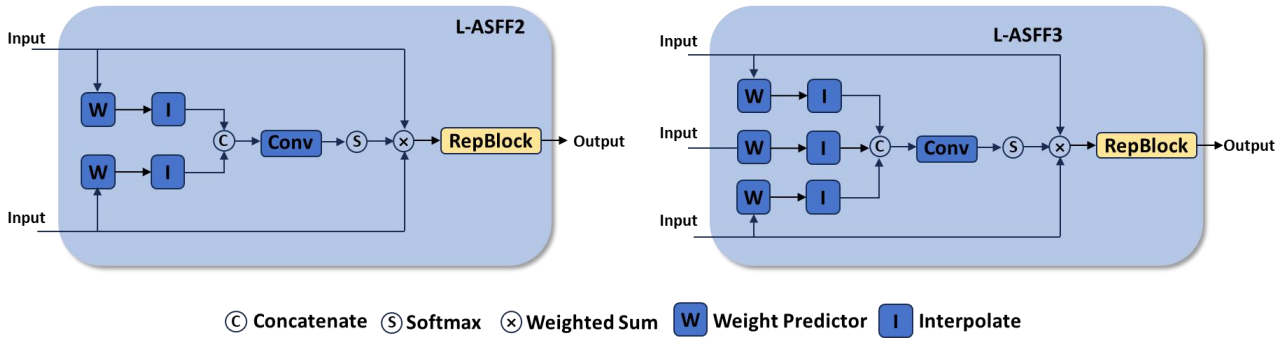

Figure 2 Architecture of the proposed Lightweight ASFF modules. L-ASFF2(left) Computes Global Fusion Weights for Two Input Feature Maps Using Pooled 1×1 Convolutions, Upsamples these Weight Maps to the Target Resolution, Applies Per-Pixel Weighted Summation, and Refines the Result with a CSP-Style Residual Block. **L-ASFF3** (right) Extends the Same Pipeline to Three Input Scales

3.3 ASFF Adaptive Multi-Scale Fusion

To address the problem of significant size differences in fire targets, we introduce lightweight Adaptive Spatial Feature Fusion (Lightweight ASFF) modules in the hybrid encoder to fully utilize features at different scales. The ASFF module can automatically learn the optimal fusion method for different scale features at each spatial location, reducing interference from inconsistent features. According to the number of input layers, we define two types of ASFF modules: ASFF-2 for fusing two scale features, and ASFF-3 for fusing three scale features. The detailed architectures of our designed lightweight ASFF-2 and ASFF-3 modules are illustrated in Figure 2.

Lightweight ASFF-2 module: The inputs are high-level feature A (lower resolution) and mid-level feature B (higher resolution, upsampled to the same size as A). To reduce computational complexity, we adopt a lightweight weight prediction strategy: first perform global average pooling on each input feature separately, then compress to 4 dimensions through 1×1 convolution to obtain global context descriptions A' and B'; then upsample A' and B' back to the original feature map size and concatenate in the channel dimension, generating a 2-channel weight map $\mathbf{W} = (W_A, W_B)$ through a 1×1 convolution. Apply Softmax normalization to $\mathbf{W}$ in the channel dimension so that the sum of the two weights at each location equals 1. Finally, multiply element-wise with corresponding scale features and add them to form the fused output:

$$Y(p) = W_A(p)\,A(p) + W_B(p)\,B(p), \quad \forall\, p \in \text{Spatial}. \tag{1}$$

where p represents pixel positions on the feature map. This design of global pooling plus weight prediction significantly reduces computational overhead while maintaining the effect of adaptive fusion. The output Y of ASFF-2 then passes through a lightweight CSPRep residual block (single-layer RepVGG structure) for fusion adjustment, enhancing the robustness of fused features.

Lightweight ASFF-3 module: Extended for simultaneously fusing high (A), mid (B), and low (C) level features. The same lightweight strategy is adopted: perform global average pooling and 4-dimensional compression on the three input features separately, upsample and concatenate them, then obtain a 3-channel weight map (W_A, W_B, W_C) through convolution, and calculate the fused output after normalization:

$$Z(p) = W_A(p)\,A(p) + W_B(p)\,B(p) + W_C(p)\,C(p). \tag{2}$$

This way, three scale features participate in weighting at each location, maximally combining deep and shallow layer information. ASFF-3 also connects to a lightweight CSPRep layer for local enhancement after fusion.

In the hybrid encoder of this model, we cleverly combine ASFF-2 and ASFF-3, completing multi-scale feature fusion in two stages: First, apply ASFF-2 to the high-level P_5 and mid-level P_4 outputs from the backbone to obtain preliminarily fused top and mid-level features; then update these features separately using lightweight residual blocks. Next, further fuse the updated features with low-level features P_3 through the ASFF-3 module to generate the final multi-scale fused features. This series of operations implements a progressive multi-scale feature fusion strategy of first pairwise fusion, then three-way fusion, allowing high, mid, and low-level features to fully communicate, helping improve detection effects for fire targets of different sizes.

It is worth noting that through global average pooling and lightweight design, the computational overhead of ASFF modules is significantly reduced compared to traditional spatial convolution, with minimal parameters. Therefore, while maintaining near real-time model operation, we significantly enhance the multi-scale representation capability of features through lightweight ASFF, providing more consistent and semantically rich information for subsequent Transformer encoding.

3.4 Gated Mixture-of-Experts Transformer Encoder

Another core component of the hybrid encoder is the introduction of Transformer encoding layers with Mixture-of-Experts mechanisms. In traditional Transformer encoders, the feed-forward layer uses the same fully connected network to transform features for all positions. This "dense computation" mode may be inefficient when processing diverse inputs. We design a gated expert routing feed-forward network (MoE-FFN) that allows different feature tokens to be processed by different sub-networks (experts), as shown in Figure 3. This approach improves representation flexibility and model capacity while controlling computational overhead through sparse activation.

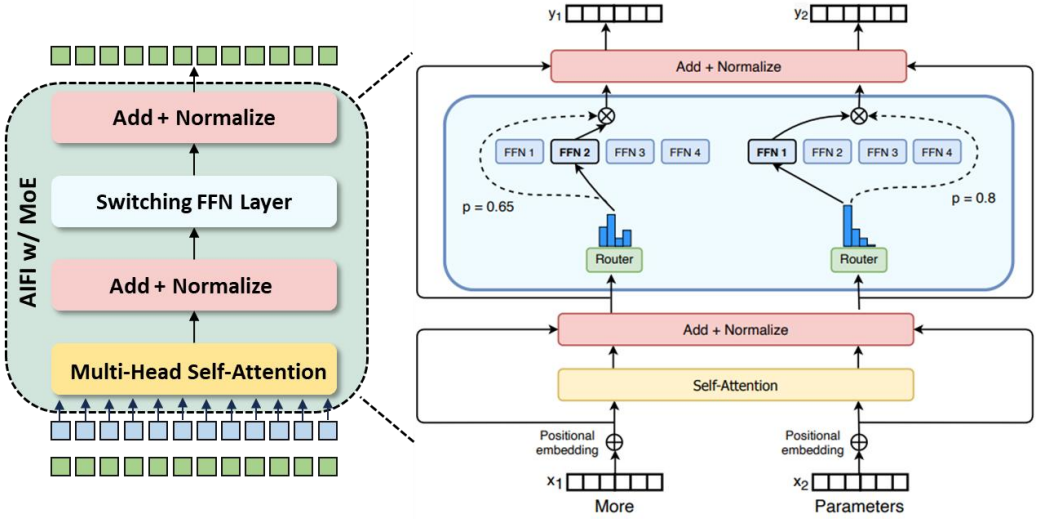

Figure 3 Detailed Architecture of the AIFI w/ MoE Module

Specifically, we retain the Multi-Head Self-Attention layer in the Transformer encoder for modeling correlations within the feature sequence. For the feature $X \in \mathbb{R}^{N \times D}$ output by attention (N is the number of tokens, D is the hidden dimension), we replace the original unified FFN layer with MoE. MoE-FFN contains one shared expert and E routable experts (all sub-layers are two-layer fully connected networks with hidden dimension d_{ff}). We also design an expert router (routing network) to determine the selected expert for each token based on input. The router is implemented as a linear layer: $\mathbf{r} = XW_r + b_r$, with output dimension E, representing the score for selecting each expert for each token. Then, we use Top-k selection (such as $k = 2$) on $\mathbf{r}$ for each token to pick the k expert indices with the highest scores and corresponding normalized weights (by applying Softmax to these k scores). This way, each token only activates k experts for computation. During actual computation, we send inputs to selected experts separately, and zero inputs make unselected experts output 0. We weight and accumulate these expert outputs according to corresponding weights to obtain the MoE-FFN transformation result for that token. Meanwhile, we add a balance loss L_{balance} to the router to encourage balanced selection frequency of all experts, avoiding overloading of certain experts.

Formally, the MoE-FFN for a token x can be expressed as:

$$\text{MoE-FFN}(x) = W_{\text{shared}}\, x + \sum_{j \in \text{Top-k}(x)} \omega_j \, f_{\text{expert},j}(x) \tag{3}$$

where $W_{\text{shared}} x$ is the output of the shared expert (serving as a common foundation for all tokens), $f_{\text{expert},j}$ represents the j-th expert sub-network, with output recorded as 0 for unselected j; ω_j is the normalized weight calculated by the router for selecting the j-th expert for x. The shared expert ensures basic capability even with poor routing, while the MoE part provides additional model capacity and diversity.

We integrate the above MoE-FFN into Transformer encoder layers, replacing the original FFN sub-layer. When the use_moe flag is enabled, the encoder layer executes: first the self-attention layer $\text{MSA}(X)$, then the MoE-FFN layer, and finally residual connection and LayerNorm normalization. If MoE is disabled, it degrades to regular FFN. It should be emphasized that during training we adopt auxiliary loss to accumulate balance losses from routing at each layer $\sum L_{\text{balance}}$; this overhead can be ignored during inference. Our implementation references OpenAI's GPT-3 Sparse MoE and Microsoft DeepSpeed MoE approaches, choosing $E = 8$ experts and setting $k = 2$ (each token activates 2 experts). In actual operation, we adopt lightweight design: the shared expert is a complete 256→1024→256 fully connected network, while the 8 routing experts are all lightweight 256→512→256 fully connected sub-networks. The computation flow for each token is: first through the shared expert (computation 1024), then activate 2 routing experts (computation 512 each), total computation approximately 2048, about 1× increase compared to the original single FFN. The model's total parameters increase by about 5× FFN parameters (1 complete shared expert + 8 half-size routing experts), but through sparse activation mechanisms, each inference still maintains small real-time computational overhead, achieving significant model capacity improvement with moderate computational increase.

3.5 Multi-Level Transformer Feature Integration

The original RT-DETR hybrid encoder only applies the Transformer encoder to the highest-level feature map (stride 32). In contrast, we consider that fire smoke also has certain semantic information at mid-level features (stride 16) with higher resolution, and may benefit from Transformer processing. Therefore, we extend the encoder to a multi-level feature integration mode: introducing Transformer encoders for multiple scale features separately and fusing their outputs again. Specifically, during HybridEncoder initialization, we can set a feature layer index list use_encoder_idx (such as including mid-level index 1 and high-level index 2), and the model will construct a separate Transformer Encoder module for each specified layer. During forward propagation, for each feature layer included, we execute its encoder, flatten 2D features into sequences, add positional encoding, send them to the encoder for self-attention and MoE-FFN transformation, then reshape results back to original feature map shape. Multi-level features enhanced by Transformer then enter the ScaleBlock multi-scale fusion module for interactive fusion. Under this design, not only do the highest-level features obtain global relationship modeling, but mid-level features can also benefit from Transformer processing, while absorbing information from other layers during fusion, further improving small target detection effects.

It should be noted that introducing multi-level Transformers brings certain computational cost increases, but we can control total costs by reducing the depth of each layer's Transformer (such as 1 encoder layer each). Additionally, RT-DETR's decoder itself supports dynamic layer number adjustment for speed control, so our model can still flexibly balance speed and accuracy during deployment.

In summary, our improved RT-DETR model fuses the advantages of convolution and Transformer in the encoder part: convolution provides local perception and enhances multi-scale representation through ASFF, ECA, etc., while Transformer introduces global dependencies and gated expert mechanisms to enhance modeling capability. The decoder part continues RT-DETR's design, using multi-layer multi-head attention to iteratively optimize queries and output detection results, with each layer having auxiliary detection heads for training. The model's training loss includes detection loss (classification, bounding box regression) and auxiliary balance loss for MoE routing, with total objective function $L = L_{\text{det}} + \lambda L_{\text{balance}}$, where λ is the weight. Through the above improvements, we expect the model to more accurately detect fire and smoke targets in drone imagery, with specific performance improvements to be verified in experiments in the next section.

4 EXPERIMENTAL DESIGN AND EVALUATION

4.1 Experimental Setup

Dataset: We evaluate our model using a self-collected and annotated UAV smoke fire dataset. This dataset contains wildfire flame and smoke images from various scenarios, totaling approximately 12,551 images. 70% are used for training, 15% for validation, and 15% for testing. The images are extracted from UAV aerial video frames with 1080p resolution, covering environments such as forests, grasslands, and mountainous areas, with fire conditions ranging from initial smoke to large-scale open flames. Annotations follow the COCO format, with each flame or smoke target marked by bounding boxes and categorized into two classes (fire or smoke). During training, we treat both classes as positive samples for detection (without distinguishing categories for mAP evaluation), while calculating individual class AP separately during evaluation for reference. Prior to model training, images undergo data augmentation including random scaling, cropping, and color jittering to improve the model's adaptability to fire conditions of different scales.

Training Details: We train all models under the PyTorch framework using the AdamW optimizer with an initial learning rate set to 1e-4. We first perform 2000 steps of linear warmup, followed by a linear decay strategy consistent with DINO to gradually reduce the learning rate from the initial value to the minimum value. Due to the relatively small dataset size, training employs pre-trained weight initialization: the ResNet50 backbone loads ImageNet pre-trained parameters, while the Transformer encoder components use Xavier random initialization. The Mixture of Experts (MoE) parameters are initialized with uniform distribution, and router biases are appropriately adjusted to encourage balance. Training is conducted for 70 epochs with a batch size of 64 (distributed data parallel training on two NVIDIA V100 GPUs). For the loss function, the detection branch uses Focal Loss (classification) and CIoU loss (bounding box), along with denoising training techniques from DN-DETR to stabilize convergence. The MoE routing balance loss coefficient λ is set to 0.01, which has been experimentally verified to achieve good results. During training, we observed that the auxiliary branch loss stabilizes after approximately 40 epochs, with overall convergence reaching optimal performance at epochs 50-60.

Evaluation Metrics: We adopt the standard COCO object detection metrics, specifically Average Precision (AP). The report primarily focuses on: mAP (mean AP) under IoU threshold 0.5:0.95 and mAP_{50} .under IoU=0.5. Additionally, to more intuitively reflect detection performance, we provide Precision and Recall metrics (using IoU=0.5 to determine true positives). Inference speed is measured by frames per second (FPS) on a single NVIDIA V100 GPU with batch size=1, tested at 640×640 scaled resolution. Model parameters (Million) and computational complexity (GFLOPs) are also provided as references. For dual-category (fire and smoke) detection, we calculate AP for each class but primarily evaluate overall model capability using comprehensive AP. All experiments are run multiple times and averaged to reduce random fluctuations.

Comparison Methods: We select several mainstream object detection models as baselines: (1) Two-stage representative: Faster R-CNN (ResNet50); (2) Single-stage representatives: YOLOv7-min (official version) and its standard version YOLOv7; (3) Transformer representative: original RT-DETR (Res18), as well as our implemented versions with various improvement modules removed for ablation studies. All aforementioned models are fine-tuned on the same dataset with identical training configurations to ensure fair comparison.

4.2 Overall Performance Comparison

Table 1 presents the performance comparison between our proposed model and mainstream detection models on the UAV smoke fire dataset test set. The results demonstrate that our improved RT-DETR achieves optimal performance across all metrics. Specifically, under the IoU=0.5 standard, our model achieves an mAP_{50} of 88.8%, representing approximately a 2 percentage point improvement over the original RT-DETR and surpassing the YOLOv7-min model by about 5 percentage points. For detection recall, our model achieves 87.9%, showing significant improvement compared to the original RT-DETR's approximately 86.7%. This indicates that our model reduces missed detections while not introducing additional false positives. The two-stage Faster R-CNN performs worst due to its insensitivity to small targets, achieving only about 80% mAP with recall below 80%, making it difficult to meet practical requirements.

Table 1 Detection Performance of Models on the UAV Smoke-Fire Dataset

Item	#Epochs	#Params (M)	GFLOPs	$FPS_{bs=1}$	mAP_{50}	$mAP_{0.5:0.95}$	Recall
Faster R-CNN	70	41.30M	134.38	21.27	0.804	0.507	0.792
YOLOv7-min	70	6.0M	6.5	171.0	0.832	0.575	0.828
YOLOv7	70	36.5M	51.6	62.7	0.894	0.643	0.861
RT-DETR	70	21.9M	29.7	86.9	0.868	0.612	0.867
Improved-RT-DETR	70	27.4M	37.1	71.5	0.888	0.638	0.879

Note: mAP_{50}/$mAP_{0.5:0.95}$ at IoU 0.50/0.50–0.95

In terms of speed, our model achieves approximately 71.5 FPS for single-frame inference on NVIDIA V100, far exceeding real-time requirements (30 FPS), though slightly lower than the original RT-DETR. This is mainly due to the introduction of additional convolutional fusion and expert parameters, which increase computational overhead. However, our model's speed remains significantly higher than the two-stage Faster R-CNN (only around 21 FPS). YOLOv7-min has the fastest inference speed, reaching 171 FPS, outperforming our model. This is because Transformer self-attention and MoE computations are more time-consuming on high-resolution feature maps. Compared to the standard YOLOv7, while it has higher accuracy than our model, it also increases corresponding parameters and computational load. Considering comprehensively factors such as accuracy, parameter count, GFLOPs, and speed, this accuracy and computational speed are acceptable for model deployment on small UAVs in fire monitoring scenarios where accuracy is prioritized. If TensorRT acceleration is used for Transformer computations, there is further room for speed improvement.

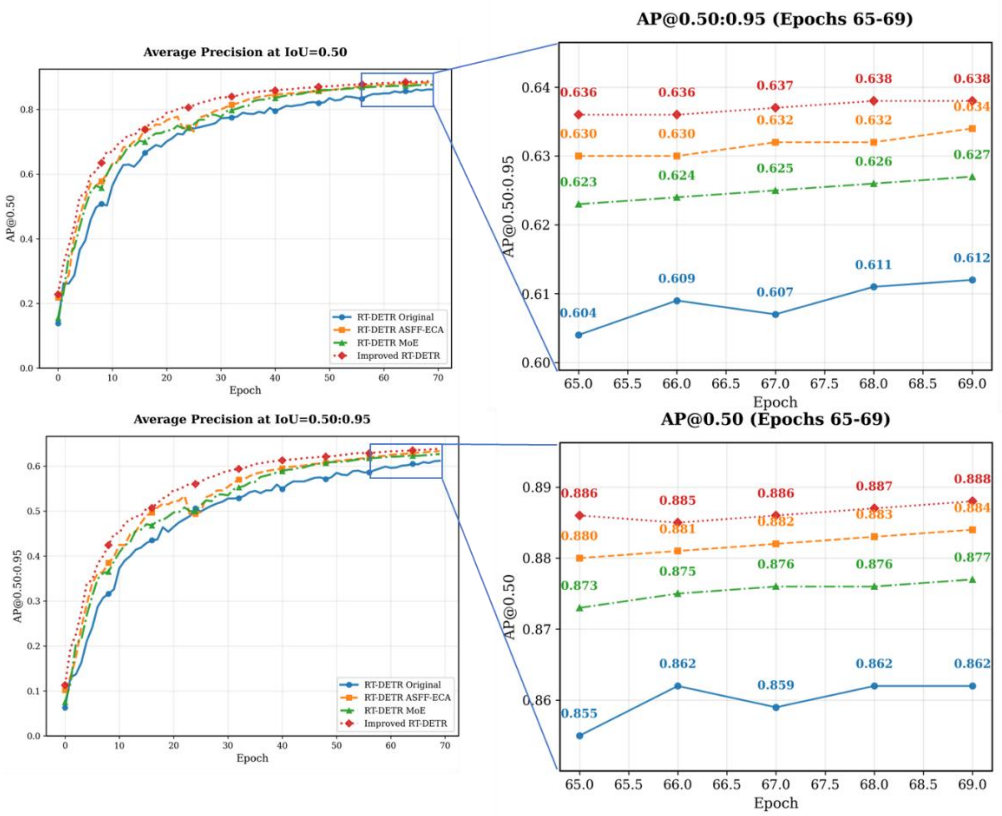

Figure 4 AP Convergence Curves (IoU = 0.50/0.50–0.95) for RT-DETR Variants

In summary, our model comprehensively outperforms the baseline RT-DETR in accuracy, particularly in detecting small flames and distant smoke columns, which is also demonstrated in the case analysis figures discussed later. Achieving such performance improvements while maintaining near real-time speed proves the effectiveness of our proposed improvement strategies (multi-scale fusion, attention enhancement, MoE expansion).

4.3 Ablation Studies

To quantify each improvement module's contribution to model performance, we designed a series of ablation experiments, with training results summarized in Figure 4. We conduct comparative analysis by progressively removing modules:

The results show that by gradually adding these modules, the model's detection accuracy steadily improves. Among them, ASFF multi-scale fusion brings the largest gain: after removing ASFF and ECA channel attention, mAP drops from 63.8% to 60.7%, a decrease of 3 percentage points, indicating that without ASFF, the model struggles to fully utilize multi-scale features, significantly degrading small target detection performance. Although ECA's contribution is less significant than ASFF, it remains non-negligible. After removing the MoE expert layer, mAP decreases by approximately 1.9 percentage points. This demonstrates that the MoE mechanism indeed provides performance improvement, validating that expert routing can enhance the model's ability to characterize different fire patterns.

Notably, the original baseline model achieves only 61.2% mAP, significantly lower than the complete model's 63.8%. This indicates that various improvements work synergistically to create the final significant enhancement. Without any component, model performance degrades to varying degrees. Particularly, ASFF fusion is crucial for information integration in small targets and complex backgrounds, serving as the key factor for our model's breakthrough over the baseline.

To intuitively demonstrate each module's role, we further compare detection results under different configurations for typical scenarios. As shown in Figure 5: in an image containing multiple distant smoke columns and multiple nearby open fire, the original model misses some smoke columns and incompletely boxes the open fire; after adding ASFF and ECA, small flames are correctly localized, proving that multi-scale fusion effectively enhances small-scale target signals; with the addition of MoE, fire boxes become more compact and accurate, and smoke is detected be slightly cause multiple experts collaborate to enhance feature response in fire regions; finally, the complete model (with MoE) has almost no missed detections in complex areas like smoke column edges, and no false detection of clouds as smoke, indicating that MoE experts further improve the model's ability to distinguish different fire appearances.

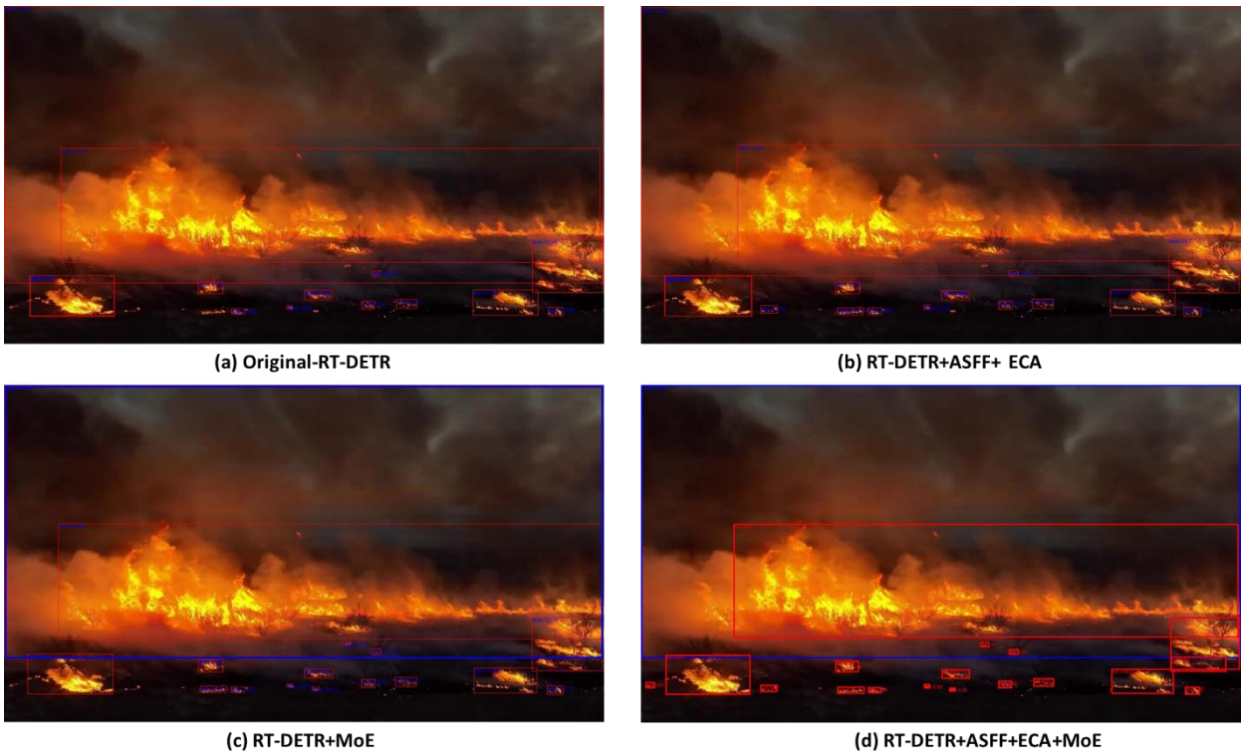

Figure 5 Comparison of Detection Effects under Different Improvement Module Configurations. (a) Original RT-DETR, Missing Some Tiny Flames; (b) +ASFF+ECA, Some Tiny Flames are Detected; (c) +MoE, Fire Target Boxes are more Accurate, and smoke is Detected; (d) ASFF+ECA+MoE full Model, all Fire Targets are Correctly Detected

5 RESULES AND DISCUSSION

5.1 Analysis of Model Improvement Effects

Based on the comprehensive experimental results above, we can clarify each improvement component's contribution to model performance enhancement:

ASFF Multi-scale Fusion: Greatly improves the model's detection capability for fire targets of different scales. Particularly in detecting distant small smoke columns, ASFF's upsampling fusion enables the model to utilize high-resolution features, significantly improving recall rate. Meanwhile, since ASFF adaptively selects feature sources at each spatial location, it reduces interference from irrelevant scale features, lowering false detection rate (Precision also improves). This is validated in both ablation experiments and visualizations. ASFF can be said to solve the insufficient cross-scale fusion problem of the original RT-DETR, and its importance aligns with conclusions from previous research on small object detection.

ECA Attention Mechanism: Helps the model better focus on discriminative features of flames and smoke. Through combined use of ECA and ASFF, the model can automatically increase channel weights for fire source highlight regions while suppressing background noise channels, playing a subtle but important role in improving detection accuracy. Although ECA's removal with ASFF only slightly decreases mAP by 1.9% in ablation studies, the localization accuracy improvement brought by ECA when used with other modules is visually apparent. This indicates ECA improves the signal-to-noise ratio of features, making the model's confidence judgment for targets more precise. Compared to SE modules, ECA requires no explicit dimensionality reduction and expansion operations, offering higher computational efficiency, making it very suitable for our real-time model.

MoE Mixture of Experts: Enhances the model's adaptability to diverse fire patterns. Since fire forms are highly varied, a single network struggles to handle all situations well, while MoE allows multiple experts to learn separately, for example, some experts focus on learning dense smoke scenarios while others specialize in open flame burning patterns. When actual input arrives, the routing network automatically selects appropriate expert combinations for processing. This mechanism effectively improves detection robustness in complex scenarios. Our model can detect partially occluded fire sources even in extreme cases (such as dense smoke obscuring open flames), which is nearly impossible with the original model. Although MoE's overall mAP improvement is less obvious than ASFF, in several difficult samples we tested, detection results with MoE enabled show significant improvement compared to when MoE is disabled. This indicates MoE's advantages mainly manifest in difficult cases—it provides the model with more capacity to characterize special situations, thereby improving the overall performance lower bound.

Multi-layer Transformer Integration: This paper primarily uses the highest-layer Transformer encoding. We attempted to simultaneously apply encoders to mid-layer features and fuse them, resulting in approximately 0.4 percentage point mAP improvement, but considering the computational cost increase of about 15%, we ultimately did not include it as a main result. However, this phenomenon merits discussion: multi-level encoding indeed further

improves performance, indicating Transformer also helps mid-layer features, but possibly due to high resolution of mid-layer features causing time-consuming attention with limited gains. Under stronger hardware or more optimized implementations, this strategy can serve as an option for balancing accuracy. Our framework design already supports flexible selection of encoding feature layers, which can be enabled as needed in the future.

5.2 Comparison with YOLO Series Methods

Although our improved model is based on RT-DETR, it's necessary to compare and discuss it with current state-of-the-art YOLO series methods. From Table 1, our model significantly outperforms YOLOv7-min in accuracy, particularly advantageous in recall rate, indicating Transformer's benefits in capturing global information and discovering hidden targets. YOLO, due to anchor mechanisms and receptive field limitations, may miss some inconspicuous smoke points. On the other hand, YOLO remains faster, mainly attributed to efficient pure CNN architecture implementation on GPUs. Therefore, in actual deployment, if pursuing ultimate speed while accepting certain missed detections, YOLOv7-min/YOLOv8-min remain good choices. However, in accuracy-prioritized scenarios (such as wildfire early warning requiring extremely low false negatives), our model provides more confident detection results.

Notably, new models like YOLOv11 also incorporate Transformer concepts (such as Decoupled Head, self-attention modules), continuously improving performance. If real-time visual Transformers further optimize speed in the future, Transformer detectors have potential to comprehensively surpass YOLO series. This research also demonstrates that by introducing excellent YOLO modules like ASFF and attention mechanisms, Transformer models' shortcomings (multi-scale and local features) can be addressed, thereby leveraging Transformer's strength in modeling global dependencies. This provides insights for future detection model design combining CNN and Transformer advantages.

5.3Model Limitations and Improvement Directions

Despite our model achieving good performance on our dataset, some limitations remain: (1) High model complexity with nearly 27.4 million parameters is oversized for some embedded platforms, hindering real-time deployment on UAV terminals. Future work could consider model pruning, distillation, or lighter backbones (such as MobileNet series) to reduce model size. (2) Our model currently only utilizes visible light image features, not yet addressing fire point detection in nighttime infrared imaging. Introducing multi-spectral data (infrared + visible light) for multi-modal fusion detection could significantly improve all-weather applicability. (3) MoE routing mechanism increases training instability; we observed that routing tends to favor certain experts in early training, requiring loss weight adjustment for convergence. Future work could explore more stable expert selection algorithms or introduce online hard example mining to make different experts' roles more distinct.

Additionally, due to our limited dataset scale, model potential may not be fully exploited. If larger-scale, more diverse UAV fire data could be collected and pre-training or semi-supervised learning employed, model performance could further improve. Some latest research directions such as video temporal information utilization, 3D convolution modeling of fire dynamics, and generative adversarial networks for synthesizing training samples are also worth trying to compensate for insufficient real data.

Overall, this research provides an effective solution for fire target detection model improvement. By combining multi-scale fusion, attention enhancement, and expert routing, we significantly improved detection accuracy while maintaining real-time performance. Looking forward, applying these strategies to more scenarios (such as urban fire monitoring, industrial accident warning) and combining with other sensor information, intelligent fire detection systems will become more robust and reliable.

6 CONCLUSION AND OUTLOOK

This paper designs an improved RT-DETR-based detection model for UAV fire detection tasks and conducts systematic experimental research on a self-built dataset. We introduce ASFF multi-scale feature fusion modules, ECA efficient channel attention mechanisms, and gated MoE mixture of experts structures into the RT-DETR model's hybrid encoder, while adopting multi-layer Transformer feature integration strategies, significantly improving the model's detection performance for flame and smoke targets of different scales. Experimental results show that compared to original RT-DETR and classic methods like YOLO and Faster R-CNN, our model has obvious advantages in detection accuracy and recall rate, particularly more accurate and reliable identification of small fire targets. Under IoU=0.5 metrics, our model achieves 88.8% mAP, improving approximately 2 percentage points over baseline with significantly reduced missed detection rate. Through ablation experiments, we quantified each improvement component's contribution, with ASFF multi-scale fusion contributing most, while ECA attention and MoE expert mechanisms also provide positive gains. Although model parameters increase somewhat, inference speed remains near real-time, meeting most UAV inspection application requirements.

Research proves that combining multi-scale fusion, attention mechanisms, and MoE expert routing can effectively enhance Transformer detectors' performance in fire monitoring domains. This provides useful reference for future development of high-precision intelligent fire monitoring systems. Looking ahead, we will further improve from the following directions: (1) Explore model lightweighting techniques such as knowledge distillation and network pruning for deployment on computation-constrained UAV platforms; (2) Expand training data including nighttime infrared fire imagery and simulated data augmentation to improve model adaptability to various conditions; (3) Extend the model to

tasks like fire spread prediction by combining video temporal information and multi-modal sensor data, achieving functionality from "seeing fire" to "predicting fire development." In the near future, with continued development of deep learning and edge computing, we have reason to expect more intelligent and efficient aerial fire monitoring systems to play key roles in forest fire prevention.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Chen Y, Zhang Y, Xin J, et al. A UAV-based forest fire detection algorithm using convolutional neural network. 2018 37th Chinese Control Conference (CCC). IEEE, 2018: 10305-10310.
[2] Haucap J, Rasch A, Stiebale J. How mergers affect innovation: theory and evidence. International Journal of Industrial Organization, 2019, 63: 283-325.
[3] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In ´ Proceedings of the IEEE/CVF International Conference on Computer Vision, 2017, 2: 2980–2988.
[4] Jocher G, Stoken A, Borovec J, et al. ultralytics/yolov5: v3. 0. Zenodo, 2020.
[5] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-ofthe-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023: 7464–7475.
[6] Mukhiddinov M, Abdusalomov A B, Cho J. A wildfire smoke detection system using unmanned aerial vehicle images based on the optimized YOLOv5. Sensors, 2022, 22(23): 9384.
[7] Zhao Y, Lv W, Xu S, et al. Detrs beat yolos on real-time object detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2024: 16965-16974.
[8] Xizhou Zhu, Weijie Su, Lewei Lu, et al. Deformable detr: Deformable transformers for end-to-end object detection. In International Conference on Learning Representations, 2020.
[9] Shilong Liu, Feng Li, Hao Zhang, et al. Dab-detr: Dynamic anchor boxes are better queries for detr. In International Conference on Learning Representations, 2021.
[10] Lv W, Zhao Y, Chang Q, et al. Rt-detrv2: Improved baseline with bag-of-freebies for real-time detection transformer. arXiv preprint arXiv:2407.17140, 2024.
[11] Liu Z, Zhang K, Wang C, et al. Research on the identification method for the forest fire based on deep learning. Optik, 2020, 223: 165491.
[12] Jiaqi Shi, Jinhu Wang, Junhui Xu, et al. Research on forest fire monitoring technology based on UAV and convolutional neural network. Advances in Applied Mathematics, 2022, 11: 3200.
[13] Jie Li, Xuanbing Qiu, Enhua Zhang, et al. Fire recognition algorithm based on convolutional neural network. Journal of Computer Applications, 2020, 40(S2): 173-177.
[14] Qiang Chen, Jian Wang, Chuchu Han, et al. Group detr v2: Strong object detector with encoder-decoder pretraining. arXiv preprint arXiv:2211.03594, 2022.
[15] Shazeer N, Mirhoseini A, Maziarz K, et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538, 2017.
[16] Fedus W, Zoph B, Shazeer N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. Journal of Machine Learning Research, 2022, 23(120): 1-39.
[17] Riquelme C, Puigcerver J, Mustafa B, et al. Scaling vision with sparse mixture of experts. Advances in Neural Information Processing Systems, 2021, 34: 8583-8595.
[18] Yuan, Jinghuil. A Margin-Maximizing Fine-Grained Ensemble Method. arXiv preprint arXiv:2409.12849, 2024.