fire

MDPI

Article

TFNet: Transformer-Based Multi-Scale Feature Fusion Forest Fire Image Detection Network

Hongying Liu [1], Fuquan Zhang [2], Yiqing Xu [3], Junling Wang [1], Hong Lu [4], Wei Wei [1] and Jun Zhu [3,*]

[1] School of Computer and Artificial Intelligence, Nanjing University of Science and Technology Zijin College, Nanjing 210023, China; liuhongying869@njust.edu.cn (H.L.); wangjunling@njfu.edu.cn (J.W.); weiwei368@njust.edu.cn (W.W.)

[2] College of Information Science and Technology & Artificial Intelligence, Nanjing Forestry University, Nanjing 210037, China; zfq@njfu.edu.cn

[3] School of Computer and Software, Nanjing Vocational University of Industry Technology, Nanjing 210023, China; yiqingxu@niit.edu.cn

[4] School of Mathematics and Information Science, Nanjing Normal University of Special Education, Nanjing 210038, China; luhong911@njts.edu.cn

* Correspondence: junzhu@niit.edu.cn

Abstract: Forest fires pose a severe threat to ecological environments and the safety of human lives and property, making real-time forest fire monitoring crucial. This study addresses challenges in forest fire image object detection, including small fire targets, sparse smoke, and difficulties in feature extraction, by proposing TFNet, a Transformer-based multi-scale feature fusion detection network. TFNet integrates several components: SRModule, CG-MSFF Encoder, Decoder and Head, and WIOU Loss. The SRModule employs a multi-branch structure to learn diverse feature representations of forest fire images, utilizing 1 × 1 convolutions to generate redundant feature maps and enhance feature diversity. The CG-MSFF Encoder introduces a context-guided attention mechanism combined with adaptive feature fusion (AFF), enabling effective multi-scale feature fusion by reweighting features across layers and extracting both local and global representations. The Decoder and Head refine the output by iteratively optimizing target queries using self- and cross-attention, improving detection accuracy. Additionally, the WIOU Loss assigns varying weights to the IoU metric for predicted versus ground truth boxes, thereby balancing positive and negative samples and improving localization accuracy. Experimental results on two publicly available datasets, D-Fire and M4SFWD, demonstrate that TFNet outperforms comparative models in terms of precision, recall, F1-Score, mAP50, and mAP50–95. Specifically, on the D-Fire dataset, TFNet achieved metrics of 81.6% precision, 74.8% recall, an F1-Score of 78.1%, mAP50 of 81.2%, and mAP50–95 of 46.8%. On the M4SFWD dataset, these metrics improved to 86.6% precision, 83.3% recall, an F1-Score of 84.9%, mAP50 of 89.2%, and mAP50–95 of 52.2%. The proposed TFNet offers technical support for developing efficient and practical forest fire monitoring systems.

Keywords: forest fire; object detection; Transformer; multi-scale feature fusion; UAV inspection

check for updates

Academic Editor: Grant Williamson

Received: 18 December 2024
Revised: 21 January 2025
Accepted: 28 January 2025
Published: 30 January 2025

Citation: Liu, H.; Zhang, F.; Xu, Y.; Wang, J.; Lu, H.; Wei, W.; Zhu, J. TFNet: Transformer-Based Multi-Scale Feature Fusion Forest Fire Image Detection Network. *Fire* **2025**, *8*, 59. https://doi.org/10.3390/fire8020059

1. Introduction

Forests are critical ecosystems on Earth and serve as habitats for diverse wildlife. However, the frequency of forest fires has been increasing annually due to global warming and human encroachment [1]. Forest fires lead to severe economic losses and casualties, as well as significant and irreversible ecological harm. Consequently, real-time forest fire monitoring is of paramount importance.

Forest fires often come with smoke, and fire detection primarily involves identifying fire points and smoke. Traditional forest fire monitoring methods primarily rely on manual ground patrols, where rangers use watchtowers or conduct inspections within forests. However, this approach may fail to detect fires promptly and can endanger the safety of rangers during large-scale fires [2]. With the development of electronics and information technology, sensor-based forest fire monitoring has emerged [3]. Sensors detect fires by measuring heat and smoke concentrations but are limited in their detection range. Remote sensing satellites provide large-scale monitoring of forest fires, but processing high-resolution remote sensing images is challenging and highly weather-dependent [4,5]. The advent of machine learning (ML) has spurred research into its application for forest fire prevention, leading to a variety of ML-based solutions [6]. For instance, Peruzzi et al. [7] utilized embedded ML models on low-power IoT devices to process audio and images for forest fire detection. Similarly, Guria et al. [8] employed ML algorithms to analyze 20 conditional factors and create forest fire probability maps for effective prevention. However, the application of machine learning in this field faces challenges such as the difficulty of obtaining forest fire data, limited real-time capabilities, poor model generalization, and high technical costs [9]. Compared to the complex conditional factor datasets required for machine learning, forest fire image data captured by drones is easier to obtain. Recent research has increasingly focused on applying deep learning techniques—such as image classification [10,11], object detection [12–14], and semantic segmentation [15]—to forest fire image analysis for fire monitoring. Drones enable 24/7 real-time monitoring of forest fires. By leveraging convolutional neural networks (CNNs), deep learning-based forest fire detection methods can directly output detection results based on learned features, allowing for timely detection and localization of fire points at the early stages of a fire [16,17].

In drone-based forest fire monitoring, drones typically begin by patrolling at high altitudes to cover larger areas. Equipped with cameras, they scan and monitor forest conditions in real time. Upon detecting fire, smoke, or suspected targets, drones lower their altitude for more accurate localization of fire points. Throughout this process, the images captured by drones are transmitted in real time to ground servers, where object detection algorithms identify fire points or smoke [18]. However, forest fire image detection using deep learning models presents several challenges: (1) Fire points in images captured at high altitudes are small, smoke is sparse, and feature extraction is difficult. (2) Deep and complex models improve feature extraction but result in large model parameters, slower processing speeds, and difficulty balancing feature extraction and model lightweighting. (3) Forest fire images have complex backgrounds, with significant occlusion and environmental noise. Smoke can easily be confused with fog, further complicating detection.

In response to these challenges, we propose TFNet, a Transformer-based multi-scale feature fusion network for detection. Specifically, TFNet addresses the difficulties posed by small fire targets, sparse smoke, and complex feature extraction by introducing a series of innovative techniques. The SRModule enhances feature extraction through a novel S-RConv structure, which improves feature diversity without significantly increasing computational costs. To effectively fuse multi-scale features, the CG-MSFF Encoder integrates a content-guided attention mechanism and adaptive feature fusion (AFF), enabling better handling of both local and global features. The Decoder optimizes the detection queries, and the WIOU Loss function improves localization accuracy by adjusting the IoU metric and balancing sample distribution. These components work together to overcome the challenges of traditional detection systems, offering a more efficient and accurate solution for forest fire detection.

This study contributes the following:

- A novel Transformer-based multi-scale feature fusion detection network (TFNet) is proposed, achieving remarkable detection accuracy on two public forest fire datasets: D-Fire and M4SFWD.
- A new SRModule is introduced for feature extraction, which replaces ResNet18's residual blocks with S-RConv.
- A content-guided multi-scale feature fusion encoder (CG-MSFF Encoder) is introduced. This encoder leverages content-guided attention to adaptively assign weights to features, effectively integrating multi-level information and extracting detailed local and global features.
- The WIOU Loss function is adopted to improve detection accuracy. By assigning differentiated weights to IoU values, WIOU balances positive and negative samples and enhances localization precision.

2. Related Work

Forests are vital ecosystems that play a crucial role in maintaining biodiversity, regulating carbon cycles and climate, protecting water sources, conserving soil fertility, and enhancing air quality. Protecting forests and promoting sustainable forest management are essential for maintaining ecological balance and ensuring long-term human development. Forest fires are among the most significant threats to forest ecosystems, causing biodiversity loss, releasing large amounts of greenhouse gases, exacerbating climate change, and leading to severe economic damage. Preventing forest fires is therefore critical for safeguarding ecosystems and human environments. Effective forest fire management involves increasing public awareness, improving fire prevention technologies, enhancing infrastructure, and adopting scientific forest management practices.

Traditional forest fire monitoring methods primarily rely on smoke detectors, thermal imaging devices, and flame sensors. However, in the open and obstructed outdoor environments typical of forests, these methods often yield significant inaccuracies. In many cases, fires are detected only after they have spread extensively [19]. To address the limitations of traditional sensor-based monitoring systems, researchers have devoted efforts to developing vision-based forest fire monitoring systems using surveillance cameras, drones, and remote sensing satellites [20].

Early vision-based forest fire monitoring methods primarily used machine learning techniques to extract features from images for distinguishing fire points from non-fire regions [21]. Numerous machine learning-based forest fire detection methods have emerged [9]. For instance, Chanthiya et al. [22] utilized SVMs to classify forest fire images into two categories, employing Euclidean distance to predict whether a query image contained flames based on test and training image comparisons. Bi et al. [23] proposed a video-based fire detection method that combined Gaussian mixture background modeling, the RGB color model, a region-growing algorithm, and an improved SVM. This method effectively identified and segmented suspected fire regions and integrated dynamic features for early fire warnings. Wahyono et al. [24] introduced a fire detection method that integrates color, motion, and shape features using machine learning. Yang et al. [25] proposed PreVM, which achieved higher detection accuracy and lower error rates by introducing novel regularization constraints. Despite their effectiveness, machine learning-based approaches for image analysis often suffer from poor generalization and are prone to overfitting. Although high-precision fire detection models can be developed using machine learning, there has been an increasing shift towards deep learning due to its superior scalability and practical applicability [26].

Deep learning techniques for detecting fire and smoke in images identify their presence and provide real-time, accurate localization. Real-time object detection can significantly reduce fire response times while minimizing human supervision costs in practical applications. Deep learning-based object detection primarily relies on convolutional neural networks (CNNs), with prominent models including the Faster R-CNN and YOLO series. Zhang et al. [27] applied Faster R-CNN for fire points and smoke detection in forest fire images, but the model achieved high detection accuracy only for simulated smoke, showing poor sensitivity to sparse smoke. Pan et al. [28] proposed a method for early fire and smoke detection using Faster R-CNN, but Faster R-CNN relies on region proposal networks (RPNs) to generate candidate regions, which is computationally expensive and time-consuming. Moreover, due to RPN limitations, the model struggles with detecting small or occluded targets, resulting in lower recall rates. Kristiani et al. [29] utilized transfer learning for real-time fire and smoke identification on intelligent edge devices, but the system's performance was significantly degraded by external noise, such as lighting variations. The YOLO series has demonstrated exceptional performance in real-time object detection tasks. Zhang et al. [30] introduced FasterNet in YOLOv5 to reduce memory consumption and increase detection speed for fire detection. Mamadaliev et al. [14] proposed a novel fire detection method for smoke based on the YOLOv8 model. Despite their strengths, YOLO models face limitations in complex environments with occlusions, small fire points, or sparse smoke.

CNN-based models often struggle to capture contextual features that are critical for accurate detection in long-distance forest fire images. To address this issue, researchers have explored attention-based models. Attention mechanisms enable models to detect small target features effectively by focusing on critical areas and capturing contextual information [31]. Han et al. [32] introduced YOLO for forest fire detection, balancing high detection accuracy with model efficiency through attention mechanisms and multi-level feature fusion. The promising results of attention mechanisms in forest fire image detection have encouraged researchers to integrate different attention mechanisms into various object detection models [33,34]. While attention mechanisms can significantly enhance feature extraction capabilities, they often require substantial computational resources, posing challenges for hardware deployment. Additionally, indiscriminate integration of attention mechanisms may lead to overfitting. Future research directions include developing innovative methods for seamlessly integrating attention mechanisms into models to improve feature extraction capabilities without significantly increasing model complexity. Balancing model accuracy and computational efficiency remains a key challenge in forest fire detection.

Our work introduces a novel approach to forest fire detection that addresses the limitations of traditional CNN and attention-based models. Unlike previous methods, which primarily rely on local feature extraction, our model leverages a Transformer-based architecture that captures long-range dependencies and contextual information. This approach significantly improves detection performance, especially for small and sparse fire points, which are often overlooked by conventional CNN models. Moreover, our method incorporates a multi-scale feature fusion mechanism through the CG-MSFF Encoder, which integrates features from different scales to enhance both local and global contexts, further improving detection accuracy. Additionally, we apply the WIOU Loss function to address the issue of sample imbalance in fire detection tasks, improving localization accuracy, particularly for small and occluded fire points.

3. Methodology

This study proposes a Transformer-based multi-scale feature fusion network (TFNet) to address the challenges of forest fire image detection. TFNet utilizes a novel feature extraction module, SRModule, as the backbone, and the outputs from the last three layers of SRModule are fed into the CG-MSFF (content-guided multi-scale feature fusion) Encoder. The CG-MSFF Encoder employs a context-guided attention mechanism and adaptive feature fusion (AFF) to weight features from different layers, effectively fusing multi-level features and extracting both local and global features. The Decoder refines these fused features, enhancing spatial resolution and capturing long-range dependencies for accurate detection. The final detection is performed by the Head, which uses a detection head to predict fire locations and classifications, with the WIOU Loss balancing positive and negative samples. The overall architecture of TFNet, illustrated in Figure 1, provides a comprehensive overview of the network's structure and how its components interact to achieve efficient and accurate forest fire image detection.

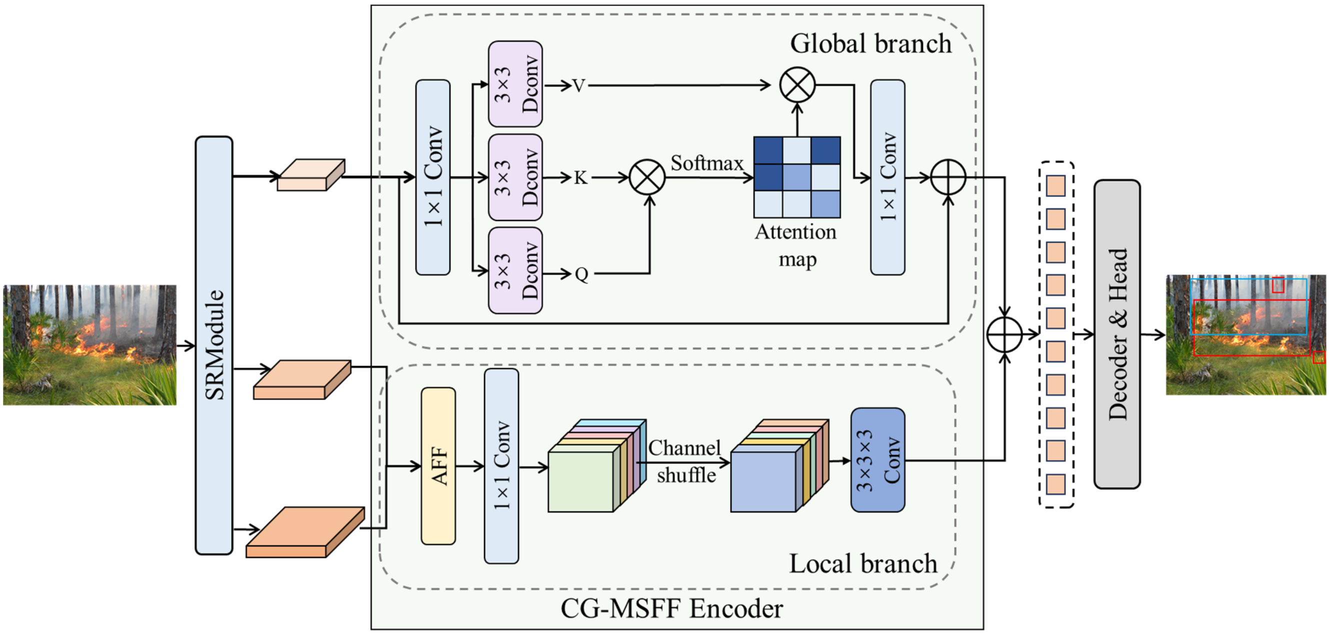

Figure 1. Overview of the TFNet model architecture.

3.1. SRModule

In the proposed architecture, the SRModule is designed to enhance the feature extraction capability while maintaining the efficiency of the network. Specifically, SRModule modifies the residual block of ResNet18 by replacing the second convolutional layer in each stage with the novel S-RConv structure.

The S-RConv plays a crucial role in enhancing feature extraction and improving the efficiency of the architecture. As shown in Figure 2, first, the input feature map undergoes an initial transformation through a 1 × 1 convolutional layer, which adjusts the channel count to twice the size of the hidden layer channels. This operation serves to capture a richer variety of features by operating across different levels of abstraction, while also setting up the data for the subsequent steps. Following this, the S-RConv performs a key operation known as the Split operation, which divides the feature map along the channel dimension into two separate branches, each containing half of the original number of channels. This division is crucial for enabling parallel processing of the input features, allowing the model to capture different aspects of the data. One of the branches is directly passed to the output without any further transformation, retaining the raw features to contribute to the final result. The second branch, however, undergoes additional processing. First, it passes through a RepConv [35] layer, which reparameterizes the convolutional weights to improve computational efficiency and reduce redundancy. Following this, the branch is processed through a 3 × 3 convolutional layer, which further refines the feature map, and then a 1 × 1 convolutional layer to adjust the output. The results of these operations are then combined with the output from the first branch, allowing the model to leverage both raw and refined features in the final output.

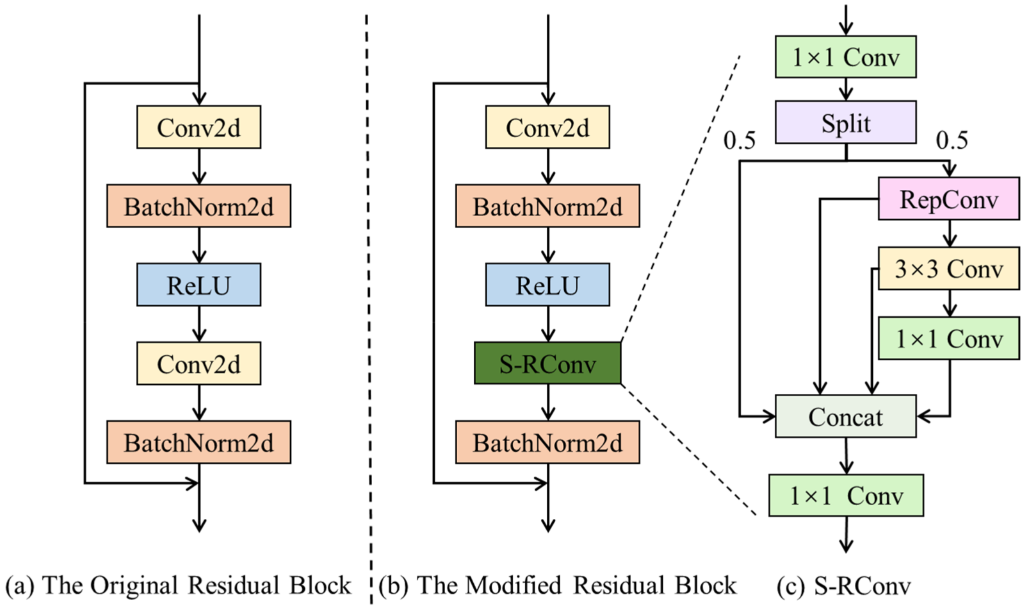

Figure 2. The S-RConv structure within the SRModule, showcasing the changes made to the residual block architecture for enhanced feature extraction.

RepConv, designed to simplify and optimize convolutions, consists of multiple branches during training. Specifically, it includes a 3 × 3 convolution followed by batch normalization (BN), a 1 × 1 convolution followed by BN, and a BN-only branch. The purpose of these branches is to reduce computational complexity by fusing the convolution and BN operations. Each branch is then converted to an equivalent 3 × 3 convolution during the fusion process. The weights of the 1 × 1 convolution are zero-padded to match the spatial dimensions of a 3 × 3 convolution, and the weights and biases from all branches are summed together to form the final convolutional weights and biases. This reparameterization step ensures that the computation is both efficient and redundant-free. During inference, the entire structure is replaced by a single 3 × 3 convolution, eliminating the need for separate convolution and BN layers and further enhancing efficiency. The resulting fused 3 × 3 convolution allows the network to maintain high performance while reducing the computational load, which is crucial for real-time applications.

3.2. CG-MSFF Encoder

The outputs of the last three stages of SRModule (F_3, F_4, and F_5) are fed into the CG-MSFF Encoder (Figure 1), which consists of a local branch and a global branch. Adaptive feature fusion (AFF), convolution, and channel shuffling operations in the local branch are used for local feature extraction. In the global branch, the self-attention mechanism is used to capture a wider range of small target information. The local branch processes F_3 and F_4, while the global branch processes F_5.

In the local branch, the feature maps F_3 and F_4 contain information at different levels of abstraction. Simply concatenating them along the channel dimension does not effectively highlight the important features and may lead to the dilution of critical information or the introduction of redundancy, which can restrict the model's detection performance. To achieve more efficient multi-scale feature fusion, we propose the adaptive feature fusion (AFF) method, as illustrated in Figure 3. By generating channel-wise attention weight matrices, AFF adaptively adjusts the contribution of each feature map, allowing for more effective fusion of features at different scales. This approach enhances the ability to emphasize important features while suppressing noise during information fusion.

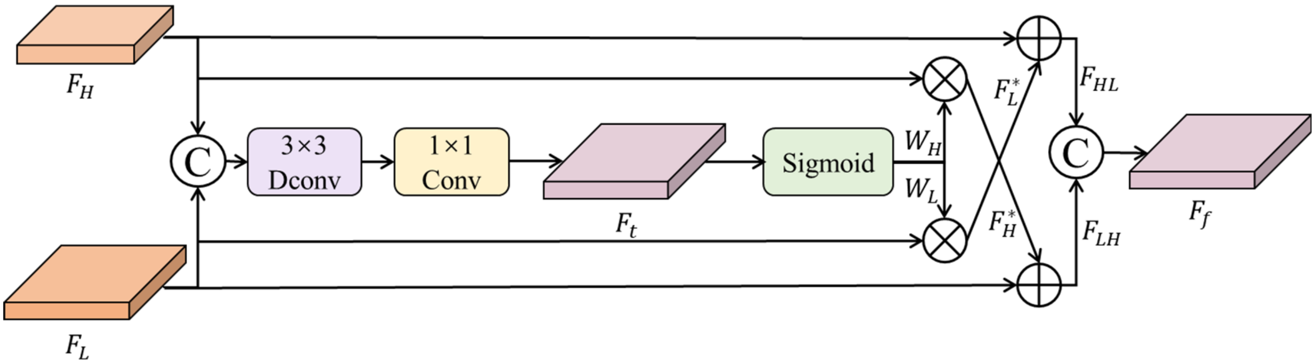

Figure 3. The AFF structure.

Specifically, the process begins by concatenating the low-level feature map F_L and the high-level feature map F_H along the channel dimension. This concatenated feature map is then passed through a depthwise separable convolution for further feature extraction, producing a new feature map F_t:

$$F_t = Conv_{1\times 1}(DConv_{3\times 3}(Concat(F_L, F_H))) \tag{1}$$

where $DConv_{3\times 3}$ denotes a depthwise convolution with a kernel size of 3, and $Concat$ represents the channel concatenation operation. Next, a sigmoid function is applied to generate two weight matrices, W_L and W_H, which will be used to adaptively adjust the contributions of the low-level and high-level features during the fusion process. These weight matrices are generated as follows:

$$x = \sigma(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

$$W_L, W_H = Split(\sigma(F_t), [C_L, C_H]) \tag{3}$$

where x represents the input to the sigmoid function, and $\sigma(x)$ is the output within the range (0, 1). The Split operation divides the feature map into two parts, with C_L and C_H representing the channel splitting parameters. W_L and W_H denote the two generated weight matrices.

The weight matrices W_L and W_H are then element-wise multiplied with the low-level feature map F_L and the high-level feature map F_H, respectively, to generate the weighted feature maps F_L^* and F_H^*:

$$F_L^* = F_L \times W_L \tag{4}$$

$$F_H^* = F_H \times W_H \tag{5}$$

After weighting, the low-level and high-level features are fused in both directions. The weighted high-level feature map F_H^* is added to the low-level feature map F_L, and the weighted low-level feature map F_L^* is added to the high-level feature map F_H. This bidirectional fusion facilitates the flow and interaction of feature information, allowing the fine-grained details from the low-level features and the semantic information from the high-level features to complement each other. This process enhances the diversity and expressiveness of the feature representation:

$$F_{LH} = F_L^* + F_H \tag{6}$$

$$F_{HL} = F_H^* + F_L \tag{7}$$

where F_{LH} and F_{HL} represent the feature maps generated after the fusion of low-level and high-level features.

Finally, the fused results are concatenated along the channel dimension and passed through a 1 × 1 convolution to perform linear combination and dimensionality reduction, yielding the final output feature map F_f:

$$F_f = Conv_{1\times1}(Concat(F_{LH}, F_{HL})) \tag{8}$$

where *Concat* denotes the channel concatenation operation. This step further enhances the expressive power of the features, improving the overall quality of the learned representations.

Furthermore, in order to enhance cross-channel interaction and facilitate information integration, a 1 × 1 convolution is first used to adjust channel dimensions. The features are then processed through a channel transformation operation, which divides features into groups, applies depthwise separable convolutions, and concatenates the output tensors along the channel dimension. A final 3 × 3 × 3 convolution extracts the following features:

$$F_{local} = Conv_{3\times3\times3}\Big(CS\Big(Conv_{1\times1}\Big(F_f\Big)\Big)\Big) \tag{9}$$

where F_{local} is the output of the local branch, $Conv_{3\times3\times3}$ is the 3 × 3 × 3 convolution, $Conv_{1\times1}$ is the 1 × 1 convolution, and CS represents the channel transformation operation.

In the global branch, after a 1 × 1 convolution, three parallel 3 × 3 convolutions are used to generate the query (Q), key (K), and value (V) tensors, each with dimensions $H \times W \times C$. Q and K are reshaped as $Q \in R^{HW\times C}$ and $K \in R^{C\times HW}$, and an attention map $ATT \in R^{C\times C}$ is calculated to reduce computational complexity:

$$ATT(Q, K, V) = V \cdot Softmax(KQ/\alpha) \tag{10}$$

where α is a learnable scaling parameter. The output F_{global} of the global branch is the following:

$$F_{global} = Conv_{1\times1} ATT(Q, K, V) + F_5 \tag{11}$$

Finally, the outputs from the local and global branches are summed to produce the CG-MSFF Encoder's final output:

$$F_{out} = F_{local} + F_{global} \tag{12}$$

3.3. Decoder and Head

The Decoder is a crucial component in the architecture, responsible for refining the initial target queries and leveraging the feature sequences produced by the CG-MSFF Encoder to predict the final object detection results. It operates by iteratively optimizing the target queries through multiple layers of self-attention and cross-attention, allowing it to capture both internal relationships within the queries and the dependencies between the queries and the encoder's feature map outputs. By progressively refining the target queries in this manner, the Decoder is able to focus on the most relevant aspects of the input feature maps, facilitating the generation of accurate class labels and bounding box coordinates. The output of the Decoder is a refined set of queries that are then passed to the Head for final prediction.

The Decoder is constructed following the structure of a classical Transformer Decoder, consisting of a stack of multiple Decoder Layers. Each Decoder Layer is composed of three key submodules: self-attention, cross-attention, and feed-forward networks (FFNs). The self-attention mechanism allows the Decoder to capture dependencies within the sequence of target queries, enabling the model to refine its understanding of the relationships between different queries. The cross-attention mechanism, on the other hand, facilitates interaction between the target queries and the encoded feature map from the encoder, ensuring that the queries are aligned with the feature map information. The feed-forward networks then further process the output of the attention mechanisms, enabling non-linear transformations and enhancing the capacity of the model to learn complex relationships within the data.

Once the Decoder has refined the queries, these queries are passed to the Head component, which maps the output of the Decoder to the final detection results. The Head consists of a simple feed-forward network, which processes the query features through two distinct branches: one for classification and one for regression. The classification branch generates the class probabilities for each query, typically through a softmax function that provides a probability distribution over all possible object classes. The regression branch, on the other hand, outputs the coordinates for the predicted bounding boxes, including parameters such as the position, width, and height of the boxes. This two-branch structure allows the Head to simultaneously handle both the classification and localization tasks, ensuring that the model can accurately predict both the category and the spatial location of objects.

3.4. WIOU Loss

WIOU Loss introduces a dynamic, non-monotonic focusing mechanism to evaluate the quality of predicted bounding boxes. It reduces the harmful gradients from low-quality predictions while enhancing overall detector performance. In TFNet, WIOU Loss is used to compute the loss between predicted and ground truth bounding boxes, improving the precision of fire point and smoke detection. The formula for WIOU Loss is the following:

$$L_{WIoU} = R_{WIoU} \cdot L_{IoU} \tag{13}$$

where R_{WIoU} is a dynamic weight factor, and L_{IoU} is the standard IoU-based loss function, defined as follows:

$$R_{WIoU} = exp(\frac{(c_x - c'_x)^2 + \left(c_y - c'_y\right)^2}{c_w^2 + c_h^2}) \tag{14}$$

$$L_{IoU} = 1 - IoU \tag{15}$$

where (c_x, c_y) and $\left(c'_x, c'_y\right)$ are the centroids of the ground truth and predicted bounding boxes, respectively; c_w and c_h are the width and height of the smallest enclosing box that contains both the predicted and ground truth boxes.

4. Experiments and Results

4.1. Datasets

The experiments in this study are conducted on two publicly available datasets, D-Fire and M4SFWD. The D-Fire [36] dataset (https://github.com/gaiasd/DFireDataset, accessed on 11 July 2024) contains 21,527 annotated images, including scenarios of fires, smoke, and the coexistence of both. For this study, we filtered the dataset to retain 9869 images featuring fire and smoke in forest environments. The M4SFWD [37] dataset (https://github.com/

Philharmy-Wang/M4SFWD, accessed on 2 August 2024) is designed for remote sensing forest fires and smoke detection. It includes 3974 annotated images of forests in various landscapes such as plains, mountains, lakes, and rivers. Both datasets are widely used for forest fire image analysis and contribute significantly to the development of effective wildfire monitoring systems [32,38]. To ensure robust model evaluation, each dataset is split into training, validation, and testing sets in a 7:2:1 ratio, as detailed in Table 1. Sample images from the D-Fire and M4SFWD datasets are shown in Figure 4.

Table 1. Details of the datasets.

Sets	D-Fire	M4SFWD	Percentage
Training	6908	2782	70%
Validation	1974	795	20%
Testing	987	397	10%

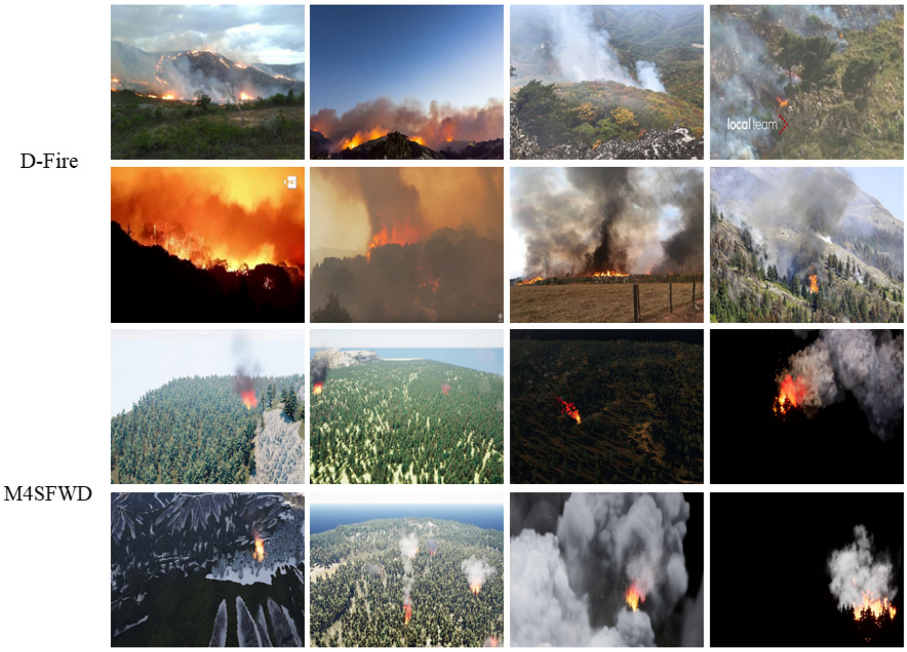

Figure 4. The sample images from the D-Fire and M4SFWD datasets.

4.2. Evaluation Metrics

All experiments are implemented using Python 3.8 on a machine equipped with an NVIDIA GeForce RTX 4090 (24 G) GPU, sourced from NVIDIA Corporation, Santa Clara, California, USA. The operating system is Ubuntu 18.04, and the PyTorch framework (version 1.11.0) is employed. The initial learning rate is set to 0.0001, the batch size to 16, and the models are trained for 200 epochs. To objectively evaluate the performance of the proposed TFNet model, the following metrics are utilized:

Precision measures the proportion of correctly identified targets among all predicted targets:

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

where TP is the number of true positives (correctly detected targets), and FP is the number of false positives (incorrect detections).

Recall measures the proportion of true targets correctly identified by the model:

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

where FN is the number of false negatives (missed targets).

F1-Score is the harmonic mean of precision and recall, offering a balanced measure of accuracy:

$$F1 - Score = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \tag{18}$$

mAP50 is the mean Average Precision (AP) computed at an IoU threshold of 0.5. A prediction is considered correct if the IoU between the predicted and ground truth bounding boxes is $\geq$0.5:

$$mAP50 = \frac{1}{N} \sum_{i=1}^{N} AP50_i \tag{19}$$

where $AP50_i$ is the Average Precision for the i-th class, calculated as follows:

$$AP50_i = \sum_t Precision_i(t) \cdot \Delta Recall_i(t) \tag{20}$$

where $\Delta Recall_i(t)$ represents the change in *Recall* at step t:

$$\Delta Recall_i(t) = Recall_i(t) - Recall_i(t-1) \tag{21}$$

mAP50–95 is the mean Average Precision over IoU thresholds ranging from 0.5 to 0.95 (step size = 0.05), providing a more comprehensive evaluation:

$$mAP50 - 90 = \frac{1}{M} \sum_{r=1}^{M} mAP50_r \tag{22}$$

where M is the total number of IoU thresholds, and $mAP50_r$ is the mAP at threshold r.

GFLOPs measures the computational complexity of the model in billions of floating-point operations per second.

$$GFLOPs = \frac{1}{10^9} FLOPs \tag{23}$$

4.3. Comparison Experiment

4.3.1. Feature Extraction Capability

To evaluate the effectiveness of the proposed TFNet model, which builds upon and improves the RT-DETR baseline, we conducted a series of experiments on the D-Fire and M4SFWD datasets. Figure 5 presents the heatmaps of feature extraction by both the baseline RT-DETR model and TFNet, highlighting the regions that are most relevant to targets. In the D-Fire dataset, TFNet demonstrated significant advantages over the baseline in its ability to focus on fire and smoke features. For example, as shown in Figure 5a, the baseline model's attention was easily distracted by complex forest backgrounds, leading to false activations, whereas TFNet effectively suppressed these distractions and concentrated on meaningful features, showcasing better robustness in handling noisy environments. Figure 5b highlights TFNet's superior sensitivity to small targets, accurately capturing fire points that the baseline model often missed. Similarly, on the M4SFWD dataset, TFNet showed considerable improvements in identifying fire points and smoke features. As seen in Figure 5c, TFNet not only detected fire points but also highlighted smoke regions, a key predictive feature for fire spread. In Figure 5d, the baseline model failed to effectively identify smoke features in dense forest areas, while TFNet consistently captured these critical indicators. These results demonstrate TFNet's ability to handle complex detection tasks in forest fire scenarios, particularly in terms of early fire and smoke detection.

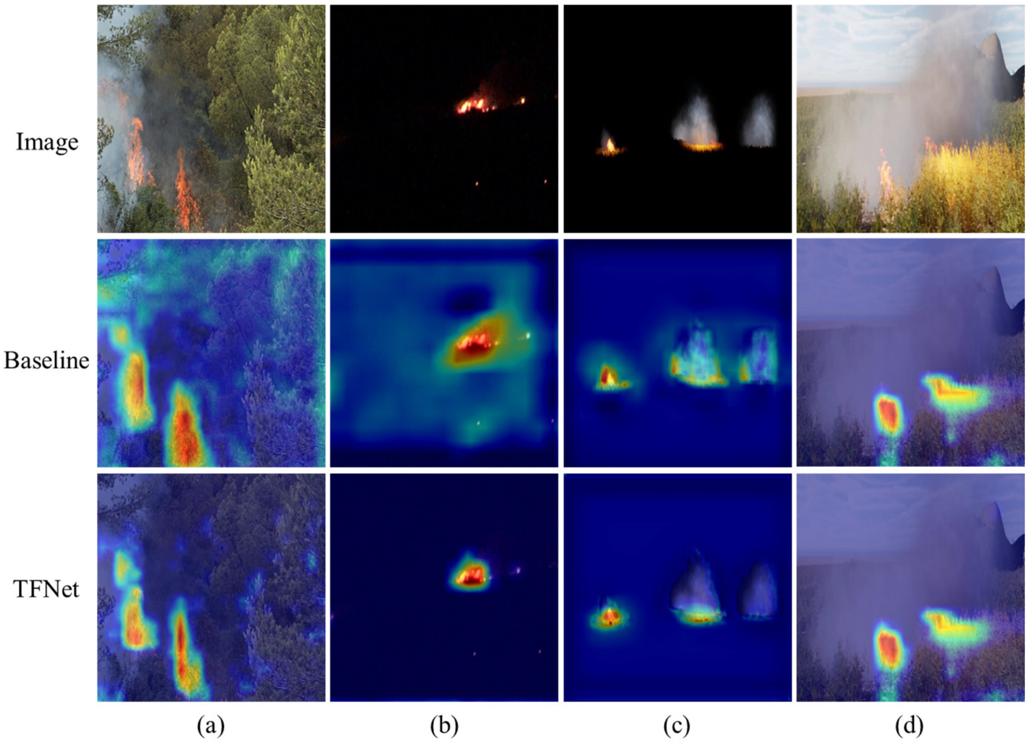

Figure 5. The heatmaps of feature extraction. (**a**,**b**) represents two samples from D-Fire and (**c**,**d**) represents two samples from M4SFWD.

4.3.2. Parameter Efficiency and Computational Complexity

In addition to feature extraction capabilities, we compared the parameter efficiency and computational complexity of TFNet with other state-of-the-art object detection models, including Faster R-CNN [39], SSD [40], Yolov5s [31], Yolov8s [14], Yolov11n [41], and DETR [42]. These factors are crucial for real-time applications like wildfire detection, where both high performance and low computational demand are essential. As shown in Table 2, TFNet strikes a good balance between model size and computational efficiency, making it ideal for resource-constrained environments.

Table 2. Parameter quantities and GFLOPs of different models.

Methods	Faster R-CNN	SSD	Yolov5s	Yolov8s	Yolov11n	DETR	TFNet
Parameters	137.1 M	26.3 M	7.0 M	11.1 M	20.1 M	41.3 M	15.4 M
GFLOPs	370.21	62.75	16.0	28.6	68.2	101.36	53.3

In terms of parameter count, TFNet has 15.4 million parameters, significantly smaller than Faster R-CNN (137.1 M) and DETR (41.3 M), which are more computationally expensive. TFNet's parameter count is higher than Yolov5s (7.0 M) and Yolov8s (11.1 M) but lower than SSD (26.3 M) and Yolov11n (20.1 M), positioning it as a competitive model for real-time applications. Despite a slightly larger parameter count than lighter models, TFNet provides better performance for detecting small or dispersed fire and smoke signals. Regarding computational complexity, TFNet achieves 53.3 GFLOPs, significantly lower than Faster R-CNN (370.21 GFLOPs) and DETR (101.36 GFLOPs), making it more efficient for real-time use. While TFNet's GFLOPs are higher than Yolov5s (16.0 GFLOPs) and Yolov8s (28.6 GFLOPs), it remains more efficient than Yolov11n (68.2 GFLOPs) and SSD (62.75 GFLOPs). This balance between model size and computational efficiency makes TFNet an excellent choice for real-time wildfire monitoring, where quick, accurate detection is crucial.

4.3.3. Loss Analysis and Model Performance

In addition to evaluating parameter efficiency, the training and testing loss variations provide crucial insights into the learning behavior and generalization capabilities of the models. TFNet and the baseline model's loss changes during training and testing serve as vital indicators of how well the models learn from the data.

The loss variations can be used to monitor training progress, adjust model structures, optimize hyperparameters, and prevent overfitting. As shown in Figure 6, which illustrates the loss changes on the M4SFWD dataset, TFNet and the baseline model exhibit distinct learning behaviors. During training, TFNet's loss drops significantly between epochs 5 and 30, showing its ability to quickly extract feature information in the early stages. Both models experience steady learning from epochs 30 to 150, but TFNet reaches a plateau earlier, starting from epoch 150, whereas the baseline model begins to stabilize only after epoch 175. This indicates that TFNet achieves optimal performance faster than the baseline model. During testing, both models exhibit decreasing test losses, indicating strong generalization capabilities. However, TFNet demonstrates a quicker decline in test loss, which stabilizes at a lower value than the baseline, suggesting that TFNet generalizes more effectively to unseen fire images.

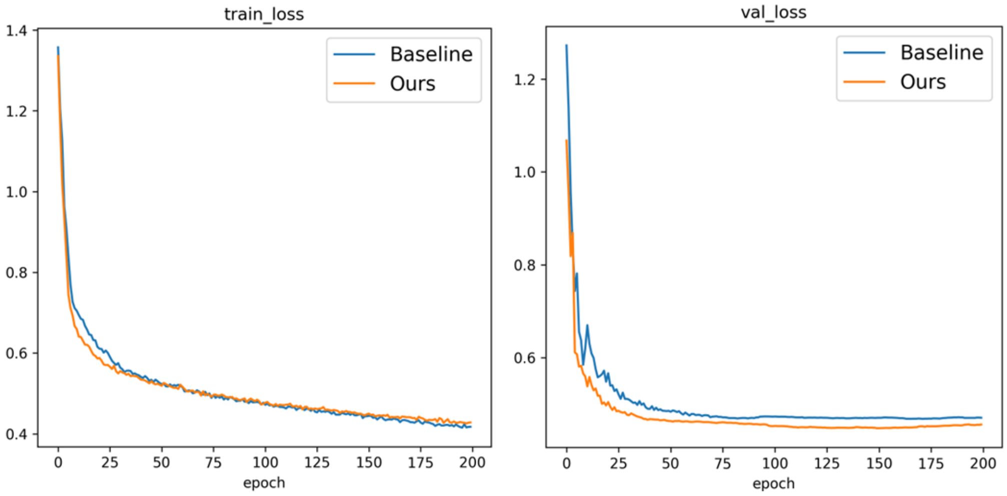

Figure 6. The training and testing loss variations.

4.3.4. Detection Performance on the D-Fire Dataset

To validate the detection performance of TFNet, we conducted extensive experiments on the D-Fire dataset and compared the results with other state-of-the-art models, as shown in Table 3. TFNet achieved an impressive precision of 81.6%, the highest among all models, demonstrating its strong ability to minimize false positives, which is crucial for real-time wildfire monitoring. The model's precision indicates that it reliably identifies true fire points and smoke without mistakenly flagging irrelevant objects. While recall for TFNet was slightly lower at 74.8% compared to Faster R-CNN's 78.2%, this difference reflects the trade-off between precision and recall. Faster R-CNN's higher recall comes from its broader target detection approach, which often leads to more false positives. In contrast, TFNet prioritizes reducing false positives, ensuring only highly relevant targets are flagged, a key advantage in wildfire detection where false alarms can be costly. TFNet also outperformed other models in F1-Score, with a value of 78.1%, reflecting an optimal balance between precision and recall. The F1-Score captures the trade-off between detecting more targets (recall) while maintaining high detection confidence (precision). TFNet's highest F1-Score highlights its well-rounded performance. Additionally, TFNet excelled in mAP50, scoring 81.2%, demonstrating not only its ability to detect fire points and smoke but also its accurate localization. The high mAP50 indicates that predicted bounding boxes closely align with the true locations. For small target detection, TFNet achieved 46.8% mAP50–95, slightly behind Yolov11n's 47.5%, but with fewer trade-offs in precision. While Yolov11n performs better on small target detection, it sacrifices precision, whereas TFNet maintains a good balance. This capability is crucial for detecting small fire spots or smoke plumes in complex forest environments. These results demonstrate that TFNet provides reliable and efficient detection across multiple aspects, even in dynamic and challenging conditions.

Table 3. Comparison experiments of different models on the D-Fire dataset.

Models	Precision (%)	Recall (%)	F1-Score	mAP50 (%)	mAP50–95 (%)
Faster R-CNN	33.3	78.2	46.7	63.5	28.4
SSD	80.7	49.6	61.4	60.6	29.6
Yolov5s	77.4	72.6	74.9	76.9	43.4
Yolov8s	77.1	72.7	74.8	77.7	46.4
Yolov11m	77.1	72.8	74.9	78.1	47.5
DETR	63.9	57.6	60.6	65.3	31.6
TFNet (ours)	81.6	74.8	78.1	81.2	46.8

The confusion matrix for the D-Fire dataset, shown in Figure 7, provides an insightful analysis of the model's classification performance. The model successfully classifies fire instances 85% of the time and smoke instances 90% of the time, demonstrating strong overall accuracy in detecting these two key categories. However, misclassifications are evident in certain areas. Specifically, the background is often incorrectly classified as either smoke (36%) or fire (64%). This suggests that the model faces challenges in effectively distinguishing between the background and the fire or smoke regions, which can lead to false positives. Furthermore, there are instances where fire is mistakenly classified as background (15%), and smoke is also occasionally misclassified as background (9%). These results underscore the model's proficiency in detecting fire and smoke but also point to areas for improvement, particularly in differentiating background elements from fire or smoke.

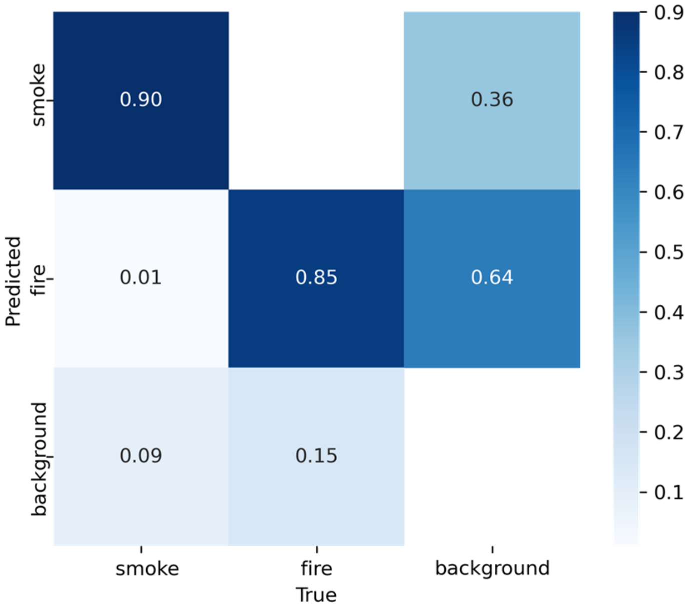

Figure 7. Confusion matrix analysis on D-Fire dataset.

The visualization analysis of the detection results further supports these methods. Figure 8 shows the sample detection results on the D-Fire dataset. Faster R-CNN, as a two-stage object detection model, often produces redundant bounding boxes and fails to handle complex forest backgrounds, leading to false negatives in the results. For example, thin smoke is missed in Figure 8a,c, and small fire points are missed in Figure 8b,d. In contrast, TFNet can consistently detect fire points and thin smoke with high accuracy even in challenging environments. Yolov11m and DETR, as representatives of one-stage object detection models and end-to-end object detection models, have higher detection accuracy than Faster R-CNN. However, they still miss small fire points and thin smoke, and their results are not as good as those of TFNet.

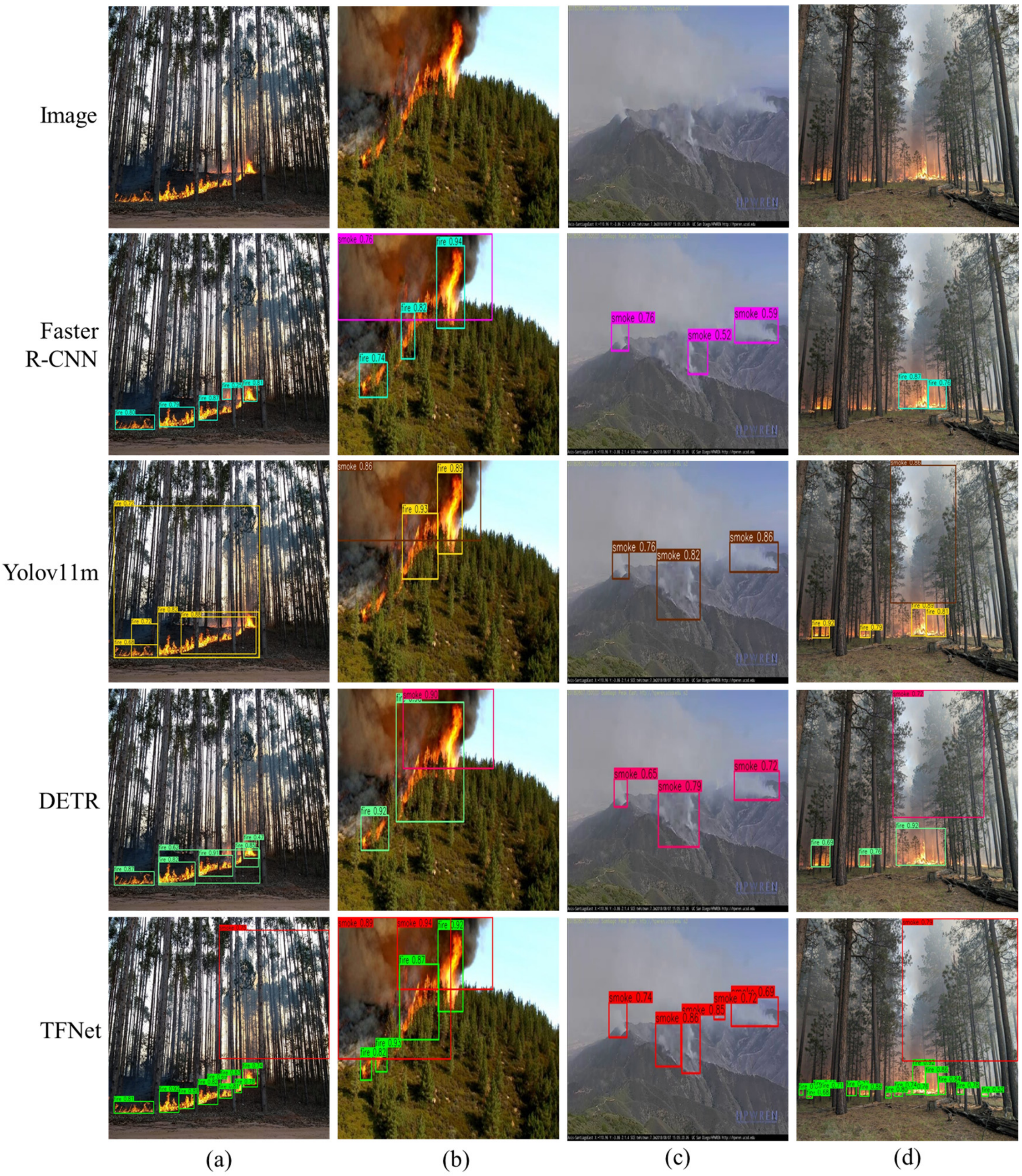

Figure 8. The sample detection results on the D-Fire dataset. (**a**–**d**) represents four different samples from D-Fire dataset.

4.3.5. Detection Performance on the M4SFWD Dataset

On the M4SFWD dataset, TFNet continued to outperform other models across most evaluation metrics, as shown in Table 4. TFNet achieved the highest precision (86.6%), significantly surpassing Faster R-CNN (49.1%) and highlighting its ability to reduce false positives. In terms of recall, TFNet scored 83.3%, slightly lower than Yolov5s (82.5%) and Yolov8s (81.6%) but higher than all other models, indicating its strong capacity for comprehensive detection. TFNet also achieved the best F1-Score (84.9%), reflecting a balanced performance in precision and recall. Furthermore, TFNet demonstrated superior performance in mAP50 (89.2%) and mAP50–95 (52.2%), underscoring its ability to handle both general and small-target detection tasks effectively. Compared to Faster R-CNN and DETR, which struggled with small fire points and sparse smoke, TFNet consistently provided accurate and reliable detection in complex forest environments.

Table 4. Comparison experiments of different models on the M4SFWD dataset.

Models	Precision (%)	Recall (%)	F1-Score	mAP50 (%)	mAP50–95 (%)
Faster R-CNN	49.1	77.5	60.1	66.2	29.0
SSD	85.1	62.2	71.9	64.8	30.8
Yolov5s	85.8	82.5	74.8	84.1	47.3
Yolov8s	84.1	81.6	82.8	87.7	49.5
Yolov11m	83.2	82.4	82.8	86.9	50.7
DETR	76.8	52.7	62.5	77.5	36.4
TFNet (ours)	86.6	83.3	84.9	89.2	52.2

The confusion matrix for the M4SFWD dataset, shown in Figure 9, shows that fire instances are correctly classified 94% of the time, with only 6% misclassified as background. Smoke instances are correctly identified 92% of the time, but 7% of smoke instances are misclassified as background and 1% as fire. In total, 30% of the background instances were misclassified as fire and 70% as smoke. These results suggest that the model is highly accurate in detecting fire and smoke but faces challenges in distinguishing background from fire and smoke, particularly with the background being misclassified as fire or smoke.

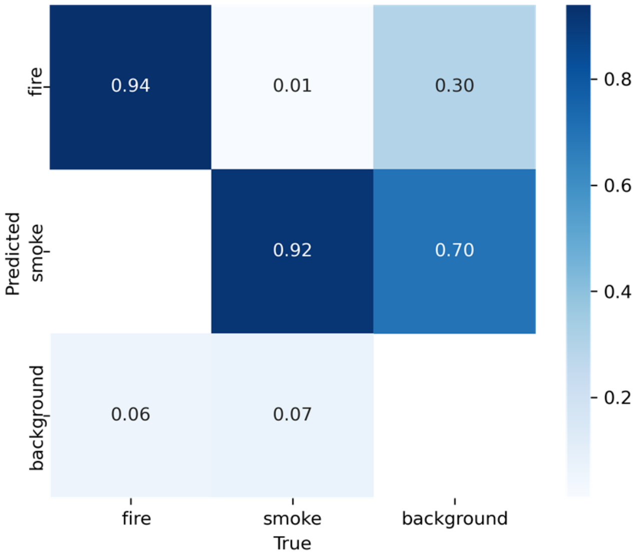

Figure 9. Confusion matrix analysis on M4SFWD dataset.

Similarly, Figure 10 presents the sample detection results on the M4SFWD dataset. Faster R-CNN and DETR are not sensitive to small fire points, which is particularly noticeable in Figure 10a. Yolov11m is not effective in detecting thin smoke, as illustrated in Figure 10c, where it struggles to correctly identify thin smoke in the presence of cloud interference. Compared to other models, TFNet can detect fire points and smoke with greater accuracy. In more complex backgrounds, such as in Figure 10a,d, TFNet can accurately detect small fire points and smoke that is partially obscured. In cases where there is an overlap between smoke and fire points, as shown in Figure 10b, TFNet can separately identify the specific locations of both smoke and fire points. In situations where the background is cluttered with clouds, as in Figure 10c, TFNet can correctly detect thin smoke.

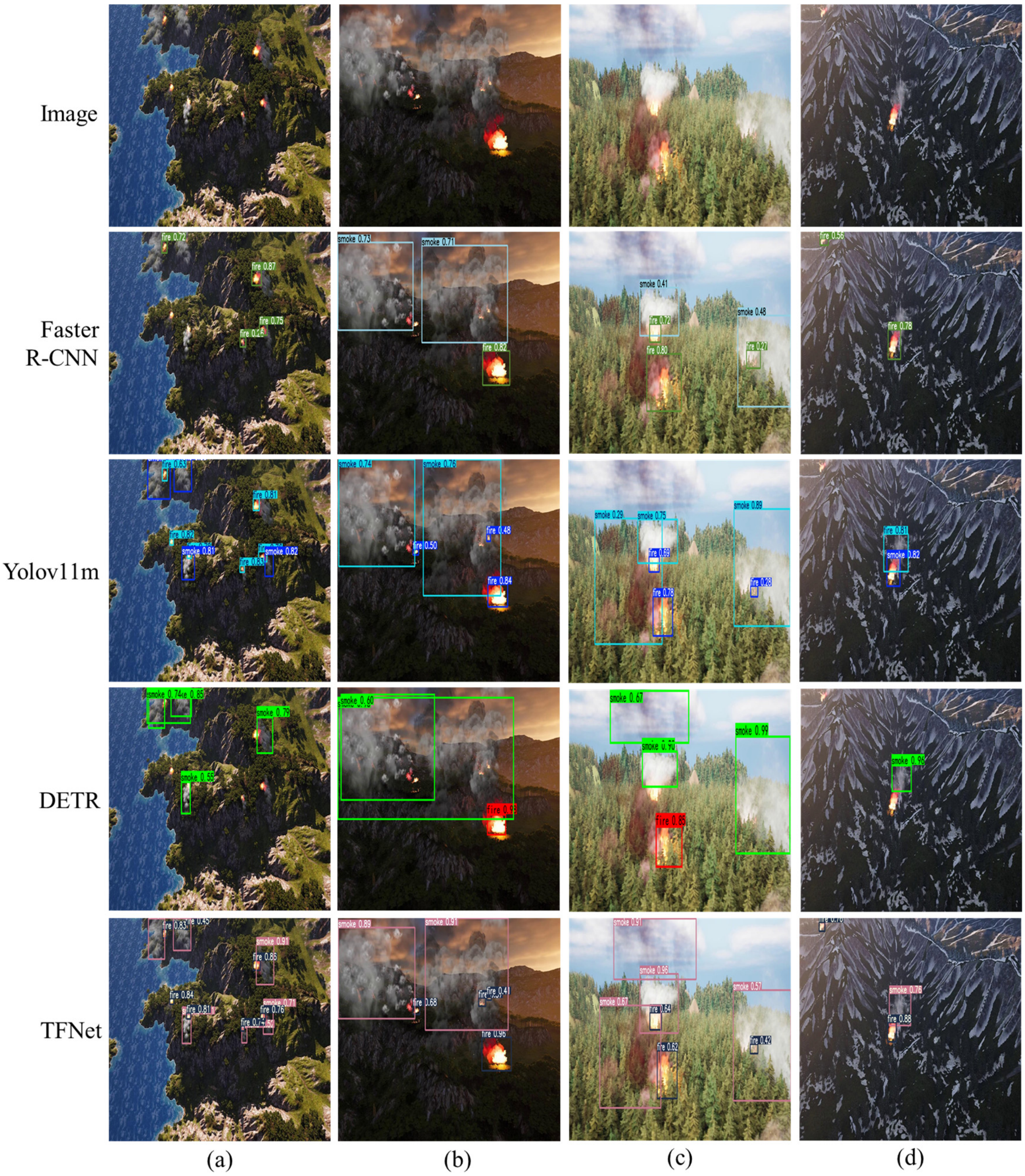

Figure 10. The sample detection results on the M4SFWD dataset. (**a**–**d**) represents four different samples from M4SFWD dataset.

4.4. Ablation Experiment

To validate the effectiveness of the SRModule, CG-MSFF Encoder, and WIoU Loss components in the TFNet model for forest fire image object detection, we conducted ablation experiments by incrementally adding these modules to the baseline RT-DETR model. For simplicity, SRModule is denoted as P1, CG-MSFF Encoder as P2, and WIoU Loss as P3. The results of these experiments on the D-Fire and M4SFWD datasets are presented in Tables 5 and 6, respectively.

Table 5. Ablation experiment results of the TFNet model on the D-Fire dataset.

Model	Precision (%)	Recall (%)	F1-Score	mAP50 (%)	mAP50–95 (%)
Baseline	79.8	73.8	76.7	79.6	45.9
Baseline + P1	80.3	74.2	77.1	80.4	46.4
Baseline + P2	81.4	73.5	77.2	80.6	46.7
Baseline + P3	81.6	74.4	77.8	80.8	46.6
Baseline + P1 + P2	81.5	74.5	77.8	81.0	46.4
Baseline + P1 + P3	79.9	75.8	77.8	80.9	46.8
Baseline + P2 + P3	83.2	72.4	77.4	80.9	46.7
Baseline + P1 + P2 + P3	81.6	74.8	78.1	81.2	46.8

Table 6. Ablation experiment results of the TFNet model on the M4SFWD dataset.

Model	Precision (%)	Recall (%)	F1-Score	mAP50 (%)	mAP50–95 (%)
Baseline	85.2	83.2	84.2	88.5	51.4
Baseline + P1	85.5	84	84.7	88.9	51.7
Baseline + P2	85.9	83.3	84.6	88.8	51.7
Baseline + P3	85.6	83.0	84.3	88.7	51.9
Baseline + P1 + P2	85.3	83.9	84.6	89.0	52.0
Baseline + P1 + P3	85.9	83.2	84.5	88.9	51.9
Baseline + P2 + P3	86.8	83	84.9	88.9	51.8
Baseline + P1 + P2 + P3	86.6	83.3	84.9	89.2	52.2

From the results on the D-Fire dataset (Table 5), it can be observed that adding SR-Module (P1) or CG-MSFF Encoder (P2) improves the F1-Score, mAP50, and mAP50–95. This demonstrates that these modules are effective in extracting features from forest fire images and fusing multi-scale information. For example, adding SRModule (Baseline + P1) improves the mAP50 from 79.6% to 80.4% and the F1-Score from 76.7% to 77.1%, while adding CG-MSFF Encoder (Baseline + P2) achieves further improvements, increasing the mAP50 to 80.6%. On the other hand, WIoU Loss (P3) slightly improves recall and mAP50–95, indicating that this loss function balances positive and negative samples and enhances localization accuracy. When combining multiple modules, such as Baseline + P1 + P2 + P3, the model achieves the best performance, with an mAP50 of 81.2%, an F1-Score of 78.1%, and an mAP50–95 of 46.8%.

On the M4SFWD dataset (Table 6), a similar trend is observed. Adding SRModule and CG-MSFF Encoder significantly improves all evaluation metrics, consistent with the findings on the D-Fire dataset. Notably, the impact of WIoU Loss is more pronounced on the M4SFWD dataset, where it contributes to higher F1-Scores and mAP50–95. For example, adding WIoU Loss (Baseline + P3) increases mAP50–95 from 51.4% to 51.9% and F1-Score from 84.2% to 84.3%. When all three modules are combined (Baseline + P1 + P2 + P3), the model achieves the best overall performance, with a precision of 86.6%, recall of 83.3%, an F1-Score of 84.9%, an mAP50 of 89.2%, and an mAP50–95 of 52.2%. These results validate the complementary contributions of SRModule, CG-MSFF Encoder, and WIoU Loss to the performance of TFNet.

To further analyze the trade-off between model performance and computational efficiency, we plotted the GFLOPs against mAP50 for various configurations of the TFNet model (Figure 11). From this scatter plot, it is evident that model parameter size and computational complexity (measured by GFLOPs) have a significant impact on detection performance (measured by mAP50). For example, the baseline model achieves an mAP50 of 79.6% on the D-Fire dataset and 88.5% on the M4SFWD dataset with a computational complexity of 57.0 GFLOPs. Adding SRModule (Baseline + P1) reduces computational complexity to 44.6 GFLOPs while improving mAP50 to 80.4% and 88.9% on the respective datasets, demonstrating that it is possible to enhance detection accuracy without significantly increasing computational overhead.

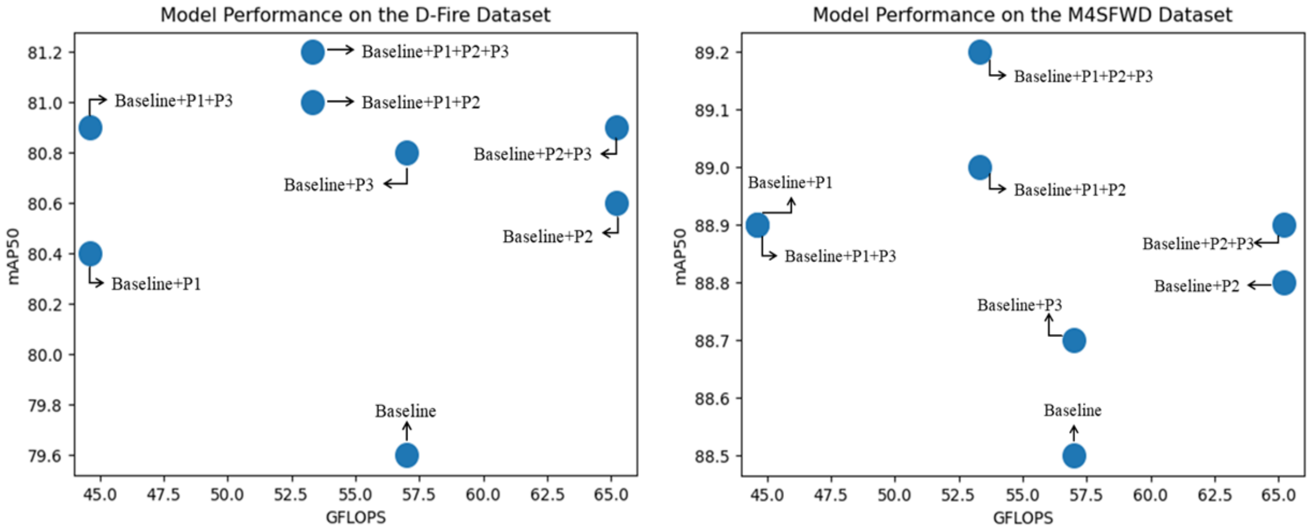

Figure 11. Scatter plot of model performance and computational complexity under different improvement strategies on different datasets.

However, certain strategies, such as adding CG-MSFF Encoder (Baseline + P2), result in increased computational complexity. For instance, Baseline + P2 raises GFLOPs to 65.2, achieving an mAP50 of 80.6% on the D-Fire dataset. While this strategy improves performance, it may not be practical for real-time applications where efficiency is critical. The combination of all three modules (Baseline + P1 + P2 + P3) achieves the best balance between performance and efficiency. This configuration maintains a relatively low computational complexity (53.3 GFLOPs) while achieving the highest mAP50 scores of 81.2% on the D-Fire dataset and 89.2% on the M4SFWD dataset. These results highlight that a thoughtful combination of optimization strategies can strike a balance between performance and computational efficiency, making the model both effective and practical for real-world applications.

5. Discussion

This research aims to tackle the challenges of small fire point targets, sparse smoke, and difficulty in feature extraction in wildfire image detection, while also improving the model's lightweight design and detection accuracy. To this end, we propose TFNet, a multi-scale feature fusion network for wildfire image detection based on Transformer. This model is an improvement upon the RT-DETR model and incorporates SRModule, CG-MSFF Encoder, and WIoU Loss modules. Through experiments on the D-Fire and M4SFWD datasets, we have validated the effectiveness of the TFNet model, which has outperformed comparison models in multiple metrics.

Compared to traditional machine learning-based wildfire image detection methods, the TFNet model demonstrates superior precision and generalization capabilities. For example, compared to the method proposed by Chanthiya et al. [22], which uses support vector machines for wildfire image classification, TFNet is able to more accurately identify fire points and smoke and precisely locate them in the image. Furthermore, in comparison with Bi et al.'s [23] video-based wildfire detection method, TFNet also shows improved detection accuracy. This improvement is primarily attributed to the model's ability to capture contextual information and small target features in the image, enabled by the Transformer structure and attention mechanism, which are superior to traditional CNN-based detection models [27,28].

The high accuracy of TFNet in wildfire image detection can be attributed to several key factors. First, the SRModule module utilizes a multi-branch structure to learn different feature representations of wildfire images, generating redundant feature maps through 1×1 convolutions. This improves the diversity of wildfire image features and effectively addresses issues such as small fire point targets, sparse smoke, and difficulty in feature extraction. Second, the CG-MSFF Encoder introduces a context-guided attention mechanism and combines AIFI with weight to effectively fuse multi-level features of wildfire images, extracting both local and global features. Third, the WIoU Loss function assigns different weights to the IoU, balancing positive and negative samples and improving the model's localization accuracy.

However, the TFNet model does have certain limitations. First, the size of the datasets used in this study is relatively small, which may affect the model's generalization ability. Second, the model's computational complexity is relatively high, and further optimization may be needed for practical applications. Third, this study did not consider the impact of environmental factors such as weather and lighting on the model's performance. Future research should explore how to improve the model's robustness in complex environments. Future improvements could focus on several areas. Expanding the dataset by collecting more wildfire images under various conditions, including different scenes, times, and weather situations, would help enhance the model's generalization ability. Additionally, exploring lighter network structures and attention mechanisms could reduce the model's computational complexity, making it more practical. Finally, addressing environmental noise through techniques such as image augmentation or the incorporation of environment-aware modules could enhance the model's robustness in complex environments.

The TFNet model proposed in this study provides a new approach for wildfire image detection and offers technical support for the development of efficient and practical forest fire monitoring systems. The model can be applied to fields such as drone inspection and remote sensing satellite monitoring, helping to detect and control wildfires in a timely manner, thereby protecting forest resources and human life and property. Furthermore, this research provides a reference for academic development in related fields, such as deep learning applications in remote sensing image processing and the use of attention mechanisms in object detection. The results also offer insights for policy making and practical operations, such as how to establish a more complete wildfire monitoring system and improve wildfire early warning and prevention capabilities.

6. Conclusions

In this study, a Transformer-based multi-scale feature fusion network for forest fire detection, TFNet, was proposed. The model was an improvement upon the RT-DETR framework by introducing SRModule, CG-MSFF Encoder, and WIoU Loss, which effectively enhanced its feature extraction capabilities and object detection accuracy. Experimental results on two public wildfire datasets, D-Fire and M4SFWD, demonstrated that TFNet achieved performance comparable to other state-of-the-art object detection models, while maintaining lower parameter counts and computational complexity. Notably, the model performed exceptionally well in small object detection. Ablation experiments further verified the effectiveness of each module in TFNet. The SRModule was shown to efficiently extract features from wildfire images, the CG-MSFF Encoder effectively fused multi-scale information, and the WIoU Loss balanced positive and negative samples, thereby improving localization precision. The proposed TFNet model provides a novel approach to forest fire detection and offers a valuable reference for future research, contributing to the protection of forest resources and the human living environment.

Author Contributions: Conceptualization, F.Z. and H.L. (Hongying Liu); software, H.L. (Hong Lu) and Y.X.; investigation, H.L. (Hong Lu); validation, H.L. (Hongying Liu) and J.Z.; formal analysis, F.Z. and Y.X.; writing—original draft preparation, H.L. (Hongying Liu) and J.W.; writing—review and editing, J.W. and W.W.; funding acquisition, J.Z. and W.W. All authors have read and agreed to the published version of the manuscript.

Funding: Natural Science Foundation of Jiangsu Province (No. BK20241070), Start-up Fund for New Talented Researchers of Nanjing Vocational University of Industry Technology (YK23-05-01), Jiangsu Province Education Science Planning Project (B-b/2024/01/163), National Computer Basic Education Research Association Project for colleges (2024-AFCEC-084), and University–Industry Collaborative Education Program (231002108075818).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be made available on request.

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

References

1. El-Madafri, I.; Peña, M.; Olmedo-Torre, N. Real-Time Forest Fire Detection with Lightweight CNN Using Hierarchical Multi-Task Knowledge Distillation. *Fire* **2024**, *7*, 392. [CrossRef]
2. Cao, L.; Shen, Z.; Xu, S. Efficient Forest Fire Detection Based on an Improved YOLO Model. *Vis. Intell.* **2024**, *2*, 20. [CrossRef]
3. Grari, M.; Yandouzi, M.; Idrissi, I.; Boukabous, M.; Moussaoui, O.; Azizi, M.; Moussaoui, M. Using IoT and ML for Forest Fire Detection, Monitoring, and Prediction: A Literature Review. *J. Theor. Appl. Inf. Technol.* **2022**, *100*, 5445–5461.
4. Wang, Y.; Wang, Y.; Xu, C.; Wang, X.; Zhang, Y. Computer Vision-Driven Forest Wildfire and Smoke Recognition via IoT Drone Cameras. *Wirel. Netw.* **2024**, *30*, 7603–7616. [CrossRef]
5. Wang, J.; Wang, Y.; Liu, L.; Yin, H.; Ye, N.; Xu, C. Weakly Supervised Forest Fire Segmentation in Uav Imagery Based on Foreground-Aware Pooling and Context-Aware Loss. *Remote Sens.* **2023**, *15*, 3606. [CrossRef]
6. Alkhatib, R.; Sahwan, W.; Alkhatieb, A.; Schütt, B. A Brief Review of Machine Learning Algorithms in Forest Fires Science. *Appl. Sci.* **2023**, *13*, 8275. [CrossRef]
7. Peruzzi, G.; Pozzebon, A.; Van Der Meer, M. Fight Fire with Fire: Detecting Forest Fires with Embedded Machine Learning Models Dealing with Audio and Images on Low Power IoT Devices. *Sensors* **2023**, *23*, 783. [CrossRef] [PubMed]
8. Guria, R.; Mishra, M.; da Silva, R.M.; Mishra, M.; Santos, C.A.G. Predicting Forest Fire Probability in Similipal Biosphere Reserve (India) Using Sentinel-2 MSI Data and Machine Learning. *Remote Sens. Appl.* **2024**, *36*, 101311. [CrossRef]
9. Sobha, P.; Latifi, S. A Survey of the Machine Learning Models for Forest Fire Prediction and Detection. *Int. J. Commun. Netw. Syst. Sci.* **2023**, *16*, 131–150. [CrossRef]
10. Islam, A.M.; Masud, F.B.; Ahmed, M.R.; Jafar, A.I.; Ullah, J.R.; Islam, S.; Shatabda, S.; Islam, A.K.M.M. An Attention-Guided Deep-Learning-Based Network with Bayesian Optimization for Forest Fire Classification and Localization. *Forests* **2023**, *14*, 2080. [CrossRef]
11. Wang, Y.; Xu, C.; Wang, Y.; Wang, X.; Ding, W. Adversarially Attack Feature Similarity for Fine-Grained Visual Classification. *Appl. Soft Comput.* **2024**, *163*, 111945. [CrossRef]
12. Wu, D.; Qian, Z.; Wu, D.; Wang, J. FSNet: Enhancing Forest-Fire and Smoke Detection with an Advanced UAV-Based Network. *Forests* **2024**, *15*, 787. [CrossRef]
13. Yang, L.; Cheng, Y.; Xu, F.; Li, B.; Li, X. Real-Time Smoke Detection in Surveillance Videos Using an Enhanced RT-DETR Framework with Triplet Attention and HS-FPN. *Fire* **2024**, *7*, 387. [CrossRef]
14. Mamadaliev, D.; Touko, P.L.M.; Kim, J.-H.; Kim, S.-C. ESFD-YOLOv8n: Early Smoke and Fire Detection Method Based on an Improved YOLOv8n Model. *Fire* **2024**, *7*, 303. [CrossRef]
15. Hu, X.; Jiang, F.; Qin, X.; Huang, S.; Yang, X.; Meng, F. An Optimized Smoke Segmentation Method for Forest and Grassland Fire Based on the UNet Framework. *Fire* **2024**, *7*, 68. [CrossRef]
16. Gonçalves, L.A.O.; Ghali, R.; Akhloufi, M.A. YOLO-Based Models for Smoke and Wildfire Detection in Ground and Aerial Images. *Fire* **2024**, *7*, 140. [CrossRef]

17. Almeida, J.S.; Jagatheesaperumal, S.K.; Nogueira, F.G.; de Albuquerque, V.H.C. EdgeFireSmoke++: A Novel Lightweight Algorithm for Real-Time Forest Fire Detection and Visualization Using Internet of Things-Human Machine Interface. *Expert Syst. Appl.* **2023**, *221*, 119747. [CrossRef]
18. Zhang, L.; Wang, M.; Ding, Y.; Wan, T.; Qi, B.; Pang, Y. FBC-ANet: A Semantic Segmentation Model for UAV Forest Fire Images Combining Boundary Enhancement and Context Awareness. *Drones* **2023**, *7*, 456. [CrossRef]
19. Yandouzi, M.; Berrahal, M.; Grari, M.; Boukabous, M.; Moussaoui, O.; Azizi, M.; Ghoumid, K.; Elmiad, A.K. Semantic Segmentation and Thermal Imaging for Forest Fires Detection and Monitoring by Drones. *Bull. Electr. Eng. Inform.* **2024**, *13*, 2784–2796. [CrossRef]
20. Titu, M.F.S.; Pavel, M.A.; Michael, G.K.O.; Babar, H.; Aman, U.; Khan, R. Real-Time Fire Detection: Integrating Lightweight Deep Learning Models on Drones with Edge Computing. *Drones* **2024**, *8*, 483. [CrossRef]
21. Rahman, M.A.; Hasan, S.T.; Kader, M.A. Computer Vision Based Industrial and Forest Fire Detection Using Support Vector Machine (SVM). In Proceedings of the 2022 International Conference on Innovations in Science, Engineering and Technology (ICISET), Chittagong, Bangladesh, 26–27 February 2022; pp. 233–238.
22. Chanthiya, P.; Kalaivani, V. Forest Fire Detection on LANDSAT Images Using Support Vector Machine. *Concurr. Comput.* **2021**, *33*, e6280. [CrossRef]
23. Bi, F.; Fu, X.; Chen, W.; Fang, W.; Miao, X.; Assef, B. Fire Detection Method Based on Improved Fruit Fly Optimization-Based SVM. *Comput. Mater. Contin.* **2020**, *62*, 199–216. [CrossRef]
24. Wahyono; Harjoko, A.; Dharmawan, A.; Adhinata, F.D.; Kosala, G.; Jo, K.-H. Real-Time Forest Fire Detection Framework Based on Artificial Intelligence Using Color Probability Model and Motion Feature Analysis. *Fire* **2022**, *5*, 23. [CrossRef]
25. Yang, X.; Hua, Z.; Zhang, L.; Fan, X.; Zhang, F.; Ye, Q.; Fu, L. Preferred Vector Machine for Forest Fire Detection. *Pattern Recognit.* **2023**, *143*, 109722. [CrossRef]
26. Park, G.; Lee, Y. Wildfire Smoke Detection Enhanced by Image Augmentation with StyleGAN2-ADA for YOLOv8 and RT-DETR Models. *Fire* **2024**, *7*, 369. [CrossRef]
27. Zhang, Q.; Lin, G.; Zhang, Y.; Xu, G.; Wang, J. Wildland Forest Fire Smoke Detection Based on Faster R-CNN Using Synthetic Smoke Images. *Procedia Eng.* **2018**, *211*, 441–446. [CrossRef]
28. Pan, J.; Ou, X.; Xu, L. A Collaborative Region Detection and Grading Framework for Forest Fire Smoke Using Weakly Supervised Fine Segmentation and Lightweight Faster-RCNN. *Forests* **2021**, *12*, 768. [CrossRef]
29. Kristiani, E.; Chen, Y.-C.; Yang, C.-T.; Li, C.-H. Flame and Smoke Recognition on Smart Edge Using Deep Learning. *J. Supercomput.* **2023**, *79*, 5552–5575. [CrossRef]
30. Zhang, D.; Chen, Y. Lightweight Fire Detection Algorithm Based on Improved YOLOv5. *Int. J. Adv. Comput. Sci. Appl.* **2024**, *15*, 809. [CrossRef]
31. Xu, H.; Li, B.; Zhong, F. Light-YOLOv5: A Lightweight Algorithm for Improved YOLOv5 in Complex Fire Scenarios. *Appl. Sci.* **2022**, *12*, 12312. [CrossRef]
32. Han, Y.; Duan, B.; Guan, R.; Yang, G.; Zhen, Z. LUFFD-YOLO: A Lightweight Model for UAV Remote Sensing Forest Fire Detection Based on Attention Mechanism and Multi-Level Feature Fusion. *Remote Sens.* **2024**, *16*, 2177. [CrossRef]
33. Kim, S.; Jang, I.; Ko, B.C. Domain-Free Fire Detection Using the Spatial–Temporal Attention Transform of the YOLO Backbone. *Pattern Anal. Appl.* **2024**, *27*, 45. [CrossRef]
34. Yang, W.; Yang, Z.; Wu, M.; Zhang, G.; Zhu, Y.; Sun, Y. SIMCB-Yolo: An Efficient Multi-Scale Network for Detecting Forest Fire Smoke. *Forests* **2024**, *15*, 1137. [CrossRef]
35. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. Repvgg: Making Vgg-Style Convnets Great Again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13733–13742.
36. de Venâncio, P.V.A.B.; Lisboa, A.C.; Barbosa, A. V An Automatic Fire Detection System Based on Deep Convolutional Neural Networks for Low-Power, Resource-Constrained Devices. *Neural Comput. Appl.* **2022**, *34*, 15349–15368. [CrossRef]
37. Wang, G.; Li, H.; Li, P.; Lang, X.; Feng, Y.; Ding, Z.; Xie, S. M4SFWD: A Multi-Faceted Synthetic Dataset for Remote Sensing Forest Wildfires Detection. *Expert Syst. Appl.* **2024**, *248*, 123489. [CrossRef]
38. Ghali, R.; Akhloufi, M.A. Deep Learning Approach for Wildland Fire Recognition Using RGB and Thermal Infrared Aerial Image. *Fire* **2024**, *7*, 343. [CrossRef]
39. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef]
40. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single Shot Multibox Detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14. Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.

41. Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv* **2024**, arXiv:2410.17725.
42. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 213–229.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.