fire

MDPI

Article

Smoke Detection Transformer: An Improved Real-Time Detection Transformer Smoke Detection Model for Early Fire Warning

Baoshan Sun [1,2,*] and Xin Cheng [1,2]

[1] School of Computer Science and Technology, Tiangong University, Tianjin 300387, China; 2331101227@tiangong.edu.cn
[2] Tianjin Key Laboratory of Autonomous Intelligence Technology and Systems, Tiangong University, Tianjin 300387, China
* Correspondence: sunbaoshan@tiangong.edu.cn

Abstract: As one of the important features in the early stage of fires, the detection of smoke can provide a faster early warning of a fire, thus suppressing the spread of the fire in time. However, the features of smoke are not apparent; the shape of smoke is not fixed, and it is easy to be confused with the background outdoors, which leads to difficulties in detecting smoke. Therefore, this study proposes a model called Smoke Detection Transformer (Smoke-DETR) for smoke detection, which is based on a Real-Time Detection Transformer (RT-DETR). Considering the limited computational resources of smoke detection devices, Enhanced Channel-wise Partial Convolution (ECPConv) is introduced to reduce the number of parameters and the amount of computation. This approach improves Partial Convolution (PConv) by using a selection strategy that selects channels containing more information for each convolution, thereby increasing the network's ability to learn smoke features. To cope with smoke images with inconspicuous features and irregular shapes, the Efficient Multi-Scale Attention (EMA) module is used to strengthen the feature extraction capability of the backbone network. Additionally, in order to overcome the problem of smoke being easily confused with the background, the Multi-Scale Foreground-Focus Fusion Pyramid Network (MFFPN) is designed to strengthen the model's attention to the foreground of images, which improves the accuracy of detection in situations where smoke is not well differentiated from the background. Experimental results demonstrate that Smoke-DETR has achieved significant improvements in smoke detection. In the self-building dataset, compared to RT-DETR, Smoke-DETR achieves a Precision that has reached 86.2%, marking an increase of 3.6 percentage points. Similarly, Recall has achieved 80%, showing an improvement of 3.6 percentage points. In terms of mAP50, it has reached 86.2%, with a 3.8 percentage point increase. Furthermore, mAP50 has reached 53.9%, representing a 3.6 percentage point increase.

Keywords: smoke detection; fire warning; object detection; improved RT-DETR; transformer-based; foreground-focus

Citation: Sun, B.; Cheng, X. Smoke Detection Transformer: An Improved Real-Time Detection Transformer Smoke Detection Model for Early Fire Warning. *Fire* **2024**, *7*, 488. https://doi.org/10.3390/fire7120488

Received: 13 November 2024
Revised: 11 December 2024
Accepted: 20 December 2024
Published: 23 December 2024

1. Introduction

As an exceptionally destructive phenomenon in nature, fire represents a significant threat to human life and safety, with its rapid and intense progression often resulting in substantial economic losses [1,2] and immeasurable ecological damage [3]. The frequency of fire occurrences and their consequences on a global scale are cause for concern. As an illustrative example, the United States records over 3000 fire-related deaths annually [4]. In 2022, Canada experienced a particularly destructive year for fires, with 5449 incidents recorded nationwide, destroying 1,610,216 hectares of land [5]. The fires caused damage not only to natural ecosystems but also severely impacted the residents' daily lives and economic activities. The essence of fire can be defined as a disaster phenomenon initiated

by a fire source and subsequently becoming uncontrollable. The rapid spread and extensive impact of fire require implementing prompt and efficacious response strategies of paramount importance.In the initial stage of a fire, the flame has not yet spread to an extent that is easily detectable. In contrast, smoke usually diffuses rapidly prior to the flame, which constitutes the most intuitive warning indication of a fire. Therefore, compared with flame detection, the rapid detection of fire smoke can issue a fire warning more quickly, thereby facilitating timely response and handling of the fire.

In the safety protection system of modern architecture, traditional technologies such as smoke detectors and heat sensors play a crucial role [6,7], particularly within indoor environments, where they can acutely capture subtle changes in the initial stages of a fire, promptly triggering alarm systems and thereby securing valuable time for personnel evacuation and firefighting operations. However, these sensor-reliant detection devices become less effective when the fire source is outdoors or in open spaces. The rapid dispersion and dilution of smoke particles and heat in open environments significantly reduce the sensitivity and accuracy of these sensors, rendering early fire warnings in such settings extremely challenging. Against this background, fire detection methods based on visual technology have emerged, offering novel perspectives in fire prevention and control. With the rapid advancement of deep learning technologies, Computer Vision capabilities in image recognition, feature extraction, and other aspects have been significantly enhanced, presenting new possibilities for smoke detection.

Similarly, smoke object detection is confronted with many challenges comparable to those encountered in certain specific detection tasks. Firstly, fire smoke exhibits a range of shades, including white, gray, and black, and may lack distinct features [8]. For example, it can be dense or sparse. Secondly, objects such as clouds and fog in outdoor backgrounds exhibit a high degree of similarity to smoke, which can result in false detection. Furthermore, the shape of smoke is indistinct, and the dimensions of smoke are not quantifiable [9]. This presents a challenge for deep learning models, which are algorithms that are highly sensitive to shape and size. This makes it difficult to accurately capture and recognize features of smoke. Therefore, despite the powerful feature extraction capabilities of deep learning models, specific methods are still required to enhance the accuracy of smoke detection. In light of the intrinsic challenges inherent to the field of smoke detection, we propose a novel real-time object detection model called Smoke Detection Transformer (Smoke-DETR), which is specifically designed for smoke detection. This model aims to address the complex and crucial task of smoke detection in an efficient and accurate manner. Furthermore, a substantial number of smoke images have been gathered from various sources, creating a multi-source smoke detection dataset containing negative samples to provide strong technical support and guarantee fire prevention. In conclusion, the principal contributions of this paper are as follows:

(1) A meticulous data curation process has yielded a diversified and high-quality dataset. The dataset extensively integrates smoke images sourced from multiple authoritative and reliable origins. This dataset encompasses not only smoke samples exhibiting multi-scale, small-area, and thin-fog morphologies but also negative samples devoid of smoke. This design thereby ensures the dataset's comprehensiveness and practicality.

(2) In the feature extraction phase, the Enhanced Channel-wise Partial Convolution (ECPConv) is employed. This method ingeniously incorporates a selection strategy to address the limitations of the Partial Convolution (PConv) [10]. Furthermore, Efficient Multi-Scale Attention (EMA) is integrated into the backbone network.This method employs channel grouping, parallel processing, and cross-spatial fusion strategies to enhance the model's capacity for image feature extraction. These improvements not only amplify the backbone network's ability to extract smoke-specific features but also to effectively reduce the model's parameter count and computational complexity.

(3) A novel foreground-focused pyramid structure called a Multi-Scale Foreground-Focus Fusion Pyramid Network (MFFPN) is presented. It leverages the multi-scale feature maps extracted by the backbone network. The aforementioned feature maps are then

subjected to a Rectangular Self-Calibration Module (RCM) in both fused and independent forms to enhance the foreground features present within the images. The RCM employs stripe convolutions in both horizontal and vertical directions, which not only enhances the learning capability of deformable objects but also reduces the parameter count in the feature fusion section.

2. Related Works

2.1. Overview of Methods for Smoke Detection

Smoke detection methods are divided into traditional vision-based smoke detection algorithms and deep learning-based smoke detection algorithms. Traditional vision-based smoke detection encompasses video-based and image-based methods. Image-based approaches focus more on smoke's color, texture, and other characteristics. Hidenori et al. [11] utilized the texture features of smoke to train an SVM for forest fire smoke recognition. H. Tian et al. [12] proposed a method to detect and separate smoke from a single image by dividing the image into quasi-smoke and quasi-background and using a dual-dictionary approach to model and separate sparse smoke. In video-based smoke detection methods, many studies rely on the motion characteristics of smoke to separate it from the background. Jia et al. [13] introduced a salient smoke detection model based on color and motion features, obtaining motion energy from video motion information to measure the saliency of suspicious smoke regions and segmenting the smoke object based on saliency. Yu et al. [14] proposed using the Lucas Kanade optical flow algorithm to calculate the optical flow of candidate regions and derived motion features from the optical flow results to distinguish smoke from other moving objects. Although video-based smoke detection can effectively extract the motion characteristics of smoke and separate it from other objects in the video, other moving objects such as swaying bags, cars driving at night, fog, or even moving people can inevitably cause false detection. Thus, it is evident that traditional Computer Vision methods typically utilize various characteristics of smoke to distinguish it from the background. Although these methods have achieved certain results in smoke detection, due to the uncertain shape and color of smoke objects and their inconspicuous features in the foreground, the manual design of feature extraction processes often contains some irrationalities. This results in certain deficiencies in effectively and comprehensively detecting smoke using traditional methods, whether based on static features such as smoke color and texture or dynamic features such as frequency, shape, or fluttering.

Deep learning approaches based on Computer Vision, renowned for their robust feature extraction capabilities, have emerged as a promising approach in smoke detection. By effectively learning the multifaceted characteristics of smoke, these methods achieve higher accuracy than traditional techniques. Common Computer Vision tasks in smoke detection encompass image classification, semantic segmentation, and object detection. Li et al. [15] introduced a novel framework that integrates traditional methods into Convolutional Neural Networks (CNNs) for wildfire smoke detection, specifically tailored for smoke image classification. This framework comprises a candidate smoke region segmentation strategy and a neural network architecture. The segmentation strategy removes complex backgrounds from wildfire smoke images, while dilated convolutions combined with DenseBlock enable the extraction of multi-scale features, thereby enhancing the accuracy of smoke classification. Wang et al. [16] proposed a hybrid network combining CNNs with Pyramid Gaussian Pooling (PGP) and a Transformer for smoke segmentation, achieving State-Of-The-Art (SOTA) performance that surpasses many previous segmentation networks. While image classification models excel in accuracy, they are limited in their ability to reflect the precise locations of smoke generation in real time. Semantic segmentation models, while capable of fine-grained image segmentation, are challenged by the ambiguous shapes of smoke, which make it difficult to annotate datasets. Furthermore, the computational demands of pixel-wise classification in semantic segmentation tasks pose practical limitations.

Object detection tasks aim to automatically identify and precisely locate objects of interest within given images or video frames. Early object detection models, such as Single Shot MultiBox Detector (SSD) [17], Region-based Convolutional Neural Network (RCNN) [18], Fast Region-based Convolutional Neural Network (Fast-RCNN) [19], and Faster Region-based Convolutional Neural Network (Faster-RCNN) [20], heavily rely on prior knowledge. For instance, SSD employs fixed anchors to predict object locations and sizes; however, anchor settings and assignment strategies often lack flexibility and Precision. R-CNN series models incur significant computational overhead in region proposal extraction and feature computation, significantly slowing detection speed. The advent of the "You Only Look Once (YOLO)" [21] model marked a significant breakthrough in real-time performance and accuracy in object detection, with the series continuously evolving and improving. The aforementioned detection methods fall under the category of anchor-based detection, requiring the generation of anchors in candidate regions followed by regression and classification tasks. However, the variable size of smoke poses a challenge in matching the span of anchor box sizes to the actual size range of smoke, introducing certain limitations in smoke detection. This variability in smoke size makes it difficult to accurately adapt the dimensions of anchor boxes, thereby hindering the effectiveness of anchor-based detection methods in identifying and locating smoke in images or video frames.

2.2. Transformer for Smoke Detection

The Transformer [22] model's triumph in the realm of Natural Language Processing (NLP) has ignited a surge of research interest in integrating the self-attention mechanism into the domain of Computer Vision (CV). The self-attention mechanism of the Transformer calculates the relevance between different parts of the input sequence. In other words, introducing self-attention into vision tasks allows the model to dynamically adjust weights across different regions. This mechanism can effectively capture long-range dependencies, thereby enhancing detection accuracy and robustness in smoke detection tasks. Consequently, the Detection Transformer (DETR) [23] model, which is based on the Transformer mechanism, has opened up a new research direction for smoke detection.

In addition to better identifying smoke objects, the DETR model boasts unique advantages. In anchor-based object detection models, post-processing steps such as Non-Maximum Suppression (NMS) are typically necessary because the model may generate multiple overlapping detection boxes that need to be merged or filtered to reduce redundancy. However, the NMS step requires time, hindering fast inference. In contrast, DETR employs the Hungarian matching algorithm to match predicted boxes with ground truth boxes, optimizing detection speed and accuracy. It achieves fast inference without NMS, significantly reducing the time required for post-processing—a capability not even achieved by the widely used Yolo series models. Recently, scholars have applied DETR-like models to smoke detection. For example, Huang et al. [24] integrated Multi-Scale Contextual Contrast Local Feature Module (MCCL) and Dense Pyramid Pooling Module (DPPM) into the DETR model, proposing an innovative DETR-based smoke detection model that significantly improved accuracy. Liang et al. [25] proposed FSH-DETR, a DETR model incorporating the Separate Single-scale Feature Interaction Module (SSFI) and the CNN-based cross-scale feature fusion module (CCFM), for multi-scale fire and smoke detection. These studies demonstrate the feasibility of Transformer-based smoke detection.

Although Transformer models can adaptively focus on smoke with no fixed shape and accelerate inference without NMS, they also come with substantial computational complexity, posing a significant challenge for real-time detection tasks. To overcome this challenge, the Real-Time Detection Transformer (RT-DETR) [26] model improves upon DETR by using fewer Transformer Encoders and efficient cross-scale feature fusion modules, significantly reducing the computational complexity of DETR-like models. Additionally, RT-DETR introduces strategies such as IoU-aware query selection, substantially enhancing the accuracy of DETR-like models. Therefore, RT-DETR is a true end-to-end object detection

model with significant application value in smoke detection tasks. It also serves as the foundational model for the subsequent research in this paper.

3. Methods

3.1. Overall Structure of Smoke-DETR

The overall structure of Smoke-DETR is depicted in Figure 1. In comparison with RT-DETR, Smoke-DETR makes improvements in the backbone network for extracting graphical features. The original ResNetBlock [27] is replaced by the ECPConvBlock, which consists of ECPConv and EMA. During the process of fusing feature maps of multiple scales, the original CCFM in RT-DETR is replaced by Multi-Scale Foreground-Focused Fusion Pyramid Network (MFFPN). When an image is input into Smoke-DETR, information on multiple scales is first extracted by the backbone, and the last three feature maps, S3, S4, and S5, of different scales are then sent into the efficient hybrid encoder. The feature map S5 is subjected to a Transformer Encoder layer called Attention-based Intra-scale Feature Interaction (AIFI), resulting in the generation of feature F5, which encapsulates deeper information. Subsequently, S3, S4, and F5 are fed into the MFFPN for feature fusion. At the end of the MFFPN, all feature maps are vectorized and concatenated. High-quality features are then selected through the IoU-aware query selection process. These selected features are fed into the decoder layer comprising deformable self-attention [28], which outputs the final detection results.

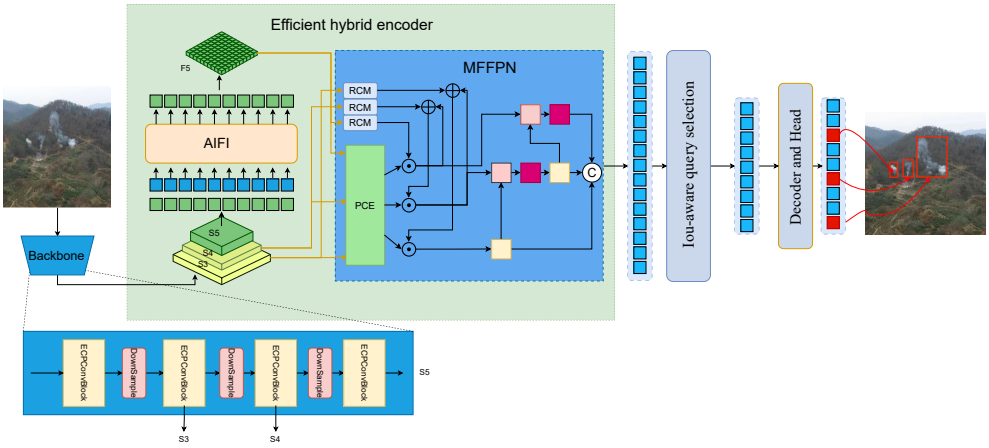


Figure 1. Smoke-DETR network architecture.

3.2. Enhanced Channel-Wise Partial Convolution

Partial Convolution (PConv) [10] is a technique that performs convolution on a subset of channels within the input feature map. For the input feature map $X \in F^{C \times H \times W}$, the computational complexity of convolution is $h \times w \times k^2 \times C_p^2$. When C_p is reduced to 1/4 of its original size, the required Floating Point Operations Per Second (FLOPs) are reduced to 1/16. This approach can effectively decrease the computational load of the model. However, this methodology overlooks the importance of various channels, thus leaving significant room for improvement. Taking this factor into account, we introduce a novel convolutional approach: Enhanced Channel-wise Partial Convolution (ECPConv).

As illustrated in Figure 2, the ECPConv architecture incorporates an initial channel screening mechanism. This mechanism evaluates the salience of individual channels and records their corresponding indices in a selection vector for subsequent processing. Subsequently, based on the indices in the selection vector, the corresponding channels are selected from the original input image. Finally, the Re-parameterized Convolution (RepConv) [29] is applied to the screened channels, and the results of the convolution are integrated back into the original image.

Algorithm A1 in Appendix A delineates the code implementation of Enhanced Channel-wise Partial Convolution. Initially, global average pooling is applied across the channels of the input feature maps to obtain a feature vector $\{x_1, x_2, x_3, ..., x_c\}$ of length C. Subsequently, we employ two fully connected layers to generate the weights for each feature channel. This process yields a weight vector $\{y_1, y_2, y_3, ..., y_c\}$. Ultimately, the weight vectors are transformed into the range 0 1 through the application of a sigmoid function, thereby indicating the relative importance of each feature channel. We order the subscripts of the feature channels according to their importance. The aforementioned result is sorted by their subscripts according to the importance of the feature channels to obtain a new vector $\{n_1, n_2, ..., n_c\}$, where n_i represents the channel number corresponding to the channel with the i-th weight. The initial K elements are retained as the basis for selecting channels to perform convolution operations, yielding $\{n_1, n_2, ..., n_k\}$, $k < c$, which is referred to as the "selection vector". According to the index in the selection vector, the corresponding channel from the original input image is selected for feature extraction, and this process is implemented through RepConv. Subsequently, the convolution result is reinserted into the original feature map in accordance with the channel index indicated in the selection vector. By generating the selection vectors prior to the convolution, this approach enhances the capacity of Partial Convolution to extract crucial channel information, thereby improving the Precision of Partial Convolution.

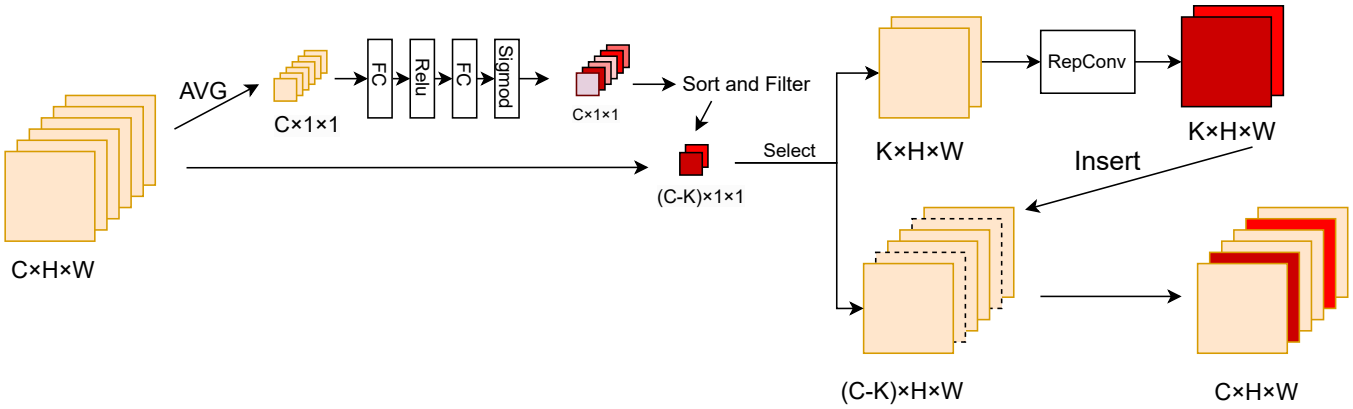


Figure 2. Illustration of Enhanced Channel-wise Partial Convolution.

3.3. Efficient Multi-Scale Attention Module

Smoke presents unique challenges in feature extraction due to its amorphous nature, variable spatial distribution, and dynamic characteristics in color and texture that fluctuate with environmental lighting conditions and smoke density. These inherent properties pose significant obstacles to the backbone network's ability to accurately extract smoke-related features. While increasing network depth could potentially enhance the learning capacity of the backbone network, this approach substantially increases the parameter count and computational overhead [30,31]. In this context, attention mechanisms emerge as an efficient alternative solution. These mechanisms, characterized by their flexible architectural properties, not only facilitate learning more discriminative feature representations but also seamlessly integrate into backbone structures. Efficient Multi-Scale Attention (EMA) [32] is an innovative attention mechanism, demonstrating remarkable effectiveness in enhancing the backbone network's learning capability through improvements upon Coordinate Attention (CA) [33]. These advantages allow us to incorporate EMA into our backbone network to further enhance the model's smoke feature recognition and learning capabilities. As illustrated in Figure 3, the operational principle of EMA encompasses three key processes: channel grouping, parallel sub-network reconstruction, and cross-spatial fusion learning.

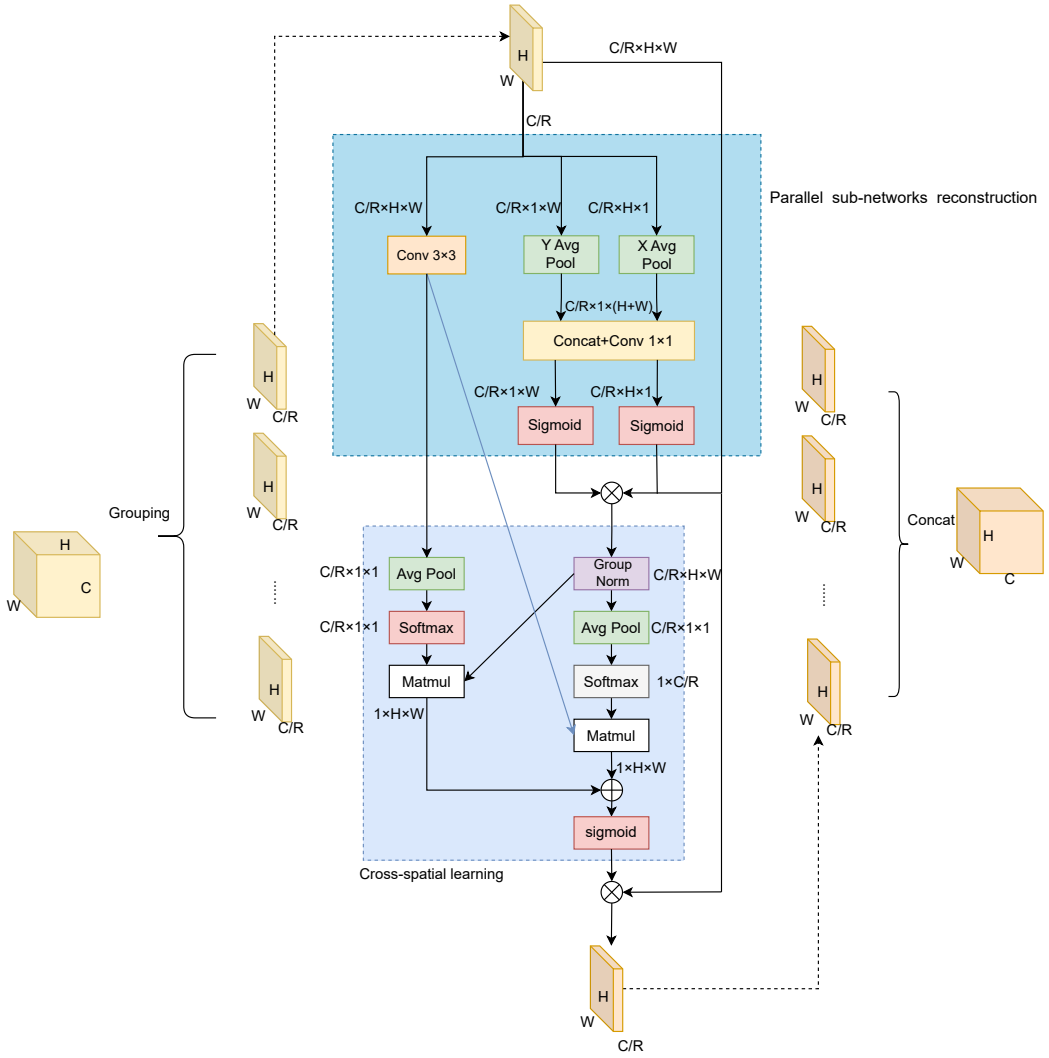


Figure 3. Processing flow of feature map by EMA.

Channel Grouping: Let the input feature map be denoted as $X \in F^{C \times H \times W}$, where C represents the number of channels, H denotes height, and W indicates width. EMA initially partitions the feature map X into R groups along the channel dimension, expressed as $X = [x_1, x_2, x_3 ... x_r]$, where each sub-feature map is represented as $x_i \in F^{C/R \times H \times W}$. In subsequent processes, the information from each sub-feature map is enhanced through parallel sub-network reconstruction and cross-spatial learning.

Parallel Sub-Network Reconstruction: EMA employs a multi-branch architecture where the input sub-feature map is divided into three independent branches. Two branches, designated as 1×1 branches, are responsible for extracting feature distributions along height and width dimensions. The third branch, termed the 3×3 branch, focuses on encoding spatial structural features. The two 1×1 branches initially perform independent horizontal and vertical average pooling to obtain feature encoding vectors in their respective directions. The horizontal and vertical average pooling processes can be formulated as follows:

$$Avg_C^H(H) = \frac{1}{W} \sum_0^W x_c(H, i) \tag{1}$$

$$Avg_C^W(W) = \frac{1}{H} \sum_0^H x_c(W, i) \tag{2}$$

where x_c represents the input feature of the c-th channel, and $x_c(H, i)$ represents the i-th value in the horizontal direction when the height is H in the c-th channel; similarly, $x_c(W, i)$ represents the i-th value in the vertical direction when the width is W in the c-th channel. The two feature encoding vectors are concatenated along the height dimension and fused through a 1×1 convolution operation. The fused result is then split and processed through sigmoid functions to generate two attention weight vectors. These attention weight vectors are subsequently aggregated with feature map X through multiplication, producing a feature map that integrates both horizontal and vertical information. Simultaneously, in the 3×3 branch, the input sub-feature map undergoes 3×3 convolution to extract spatial structural features from the original input. This approach enables EMA to not only encode inter-channel information for adjusting channel importance but also preserve precise spatial structural information within channels. Additionally, the parallel structure helps the network avoid excessive sequential processing, efficiently utilizing computational resources.

Cross-Spatial Learning: Recent Computer Vision tasks have widely adopted methods integrating spatial and channel information [34,35]. Inspired by this concept, EMA fuses the two feature maps obtained from the parallel sub-network reconstruction stage during cross-spatial learning. Specifically, the global information feature map generated by the 1×1 branch undergoes normalization. Since this normalization occurs only within each subgroup, we term it group normalization. Information is then compressed through 2D global average pooling, described by the following formula:

$$Avg_c = \frac{1}{H \times w} \sum_{i=0}^{H} \sum_{j=0}^{W} x_c(i, j) \tag{3}$$

This process enables the acquisition of global spatial information encoding. The encoding is processed through a softmax operation and undergoes matrix multiplication with the 3×3 branch's output of parallel sub-network reconstruction, generating the first spatial attention map that incorporates cross-scale spatial information. Correspondingly, the 3×3 branch output undergoes 2D global average pooling followed by a softmax operation. It then undergoes matrix multiplication with the group-normalized global information feature map, producing a second spatial attention map containing precise spatial information. These two spatial attention maps are combined and processed through a Sigmoid function to obtain the final attention map. For each sub-feature map $x_i \in F^{C/R \times H \times W}$, each channel $x_{ic} \in F^{1 \times H \times W}$ of x_i is multiplied by the corresponding group's attention map to adjust pixel information, producing the final result. Finally, the grouped sub-feature maps are reaggregated.

3.4. Improved Backbone Network Structure

The backbone network structure of Smoke-DETR is shown in Figure 4, with ECPConv and EMA as the main parts, where B represents the *batch_size*, C_i denotes the number of channels, and H and W signify the width and height of the input respectively. For the input 3-channel map, feature extraction is performed by ECPConvBlock, followed by downsampling, and the process is repeated for a total of four feature extractions with three downsamplings, where the last three extracted features will be used as inputs to the multi-scale feature module.

In the feature extraction process, ECPConv convolves the channels with high importance. While the unselected channels are not simply discarded, their information is subsequently utilized by two 1×1 convolutions of the intermediate entrainment Batch Normalization (BN) and Rectified Linear Unit (ReLu) functions. The 1×1 convolution contains less computational complexity, which aligns with our original intention of reducing the computational complexity. The original inputs are subsequently residually concatenated, and EMA enhances the results for output.

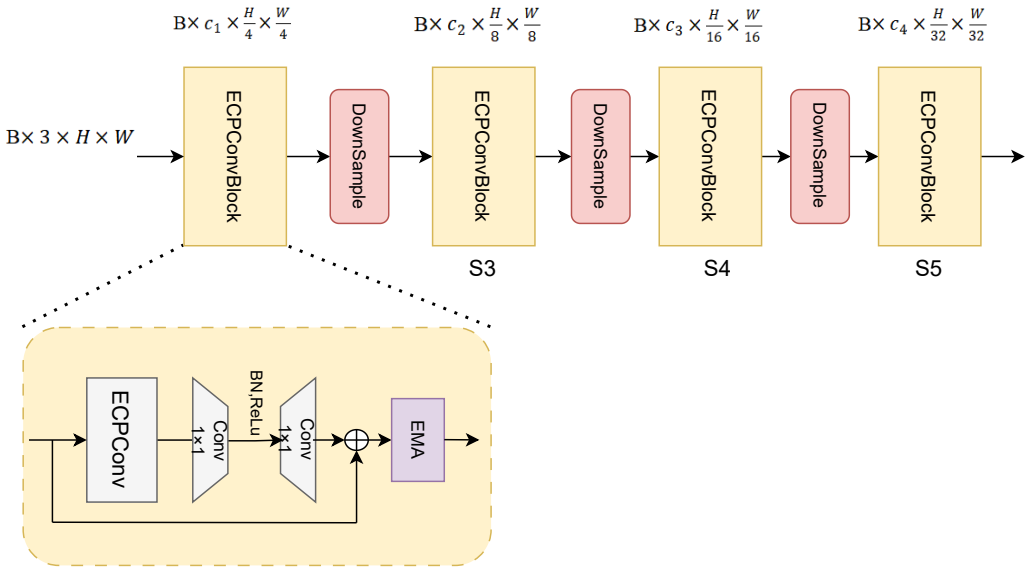


Figure 4. Improved backbone network structure of Smoke-DETR.

3.5. Multi-Scale Foreground-Focus Fusion Pyramid Network

In the context of smoke detection, it is not uncommon for objects such as clouds and fog to be mistaken for smoke, which can lead to false detection phenomena. Meanwhile, thin smoke can exhibit low discriminability from background objects, resulting in missed detections. From a visual perspective, this issue can be addressed by focusing on the foreground of smoke and enhancing the ability to distinguish it from the background. CGRSeg [36] introduces a Rectangular Self-Calibration Module (RCM), which is specifically designed for the object foreground. Based on the RCM, we have devised a Multi-Scale Foreground-Focus Fusion Pyramid Network (MFFPN) to enhance the model's capacity to focus on foreground objects at varying scales. The overall architecture is illustrated in Figure 5, which depicts two components: multi-scale feature foreground enhancement and foreground feature fusion modules. The multi-Scale feature foreground enhancement module integrates features from multiple scales. It applies the RCM for foreground-focused fusion, yielding a foreground-focused feature map incorporating multiple scales. The foreground feature fusion module employs RCM to process feature maps of varying scales, thereby generating independent Multi-Scale Foreground-Focused feature maps. These are subsequently integrated with the output of the multi-scale feature foreground enhancement module, resulting in an information feature map that encompasses both fused and independent foreground-focused maps. Ultimately, the foreground-focused feature maps of disparate scales are transformed into a vector from top to bottom, representing the output of the encoder.

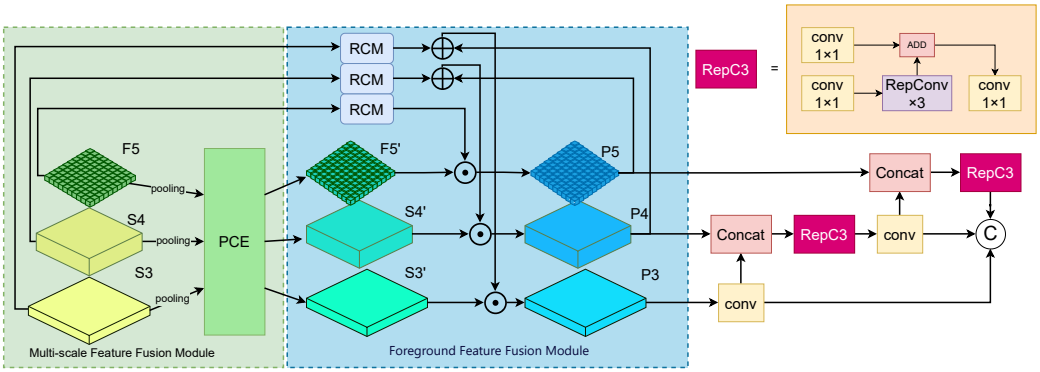


Figure 5. Overall framework of the Multi-Scale Foreground-Focus Fusion Pyramid Network.

3.5.1. Rectangular Self-Calibration Module

In MFFPN, the RCM enables the model to focus more on the foreground. The intricate architecture of the RCM is depicted in Figure 6, encompassing a Rectangular Self-Calibration Attention, a BN layer, and a Multi-Layer Perceptron (MLP) layer. The Rectangular Self-Calibration (RCA) initially captures contextual information in two dimensions using horizontal pooling and global pooling. Subsequently, this information is fused through the broadcast addition of vectors, enabling the modeling of significant foreground information. The process can be mathematically represented as follows:

$$y = Avg_C^H(F_{in}) \oplus Avg_C^w(F_{in}) \tag{4}$$

where $Avg_C^H(x)$ represents horizontal pooling of the input x, Avg_C^w represents vertical pooling of the input x, and $\oplus$ represents broadcast vector addition.

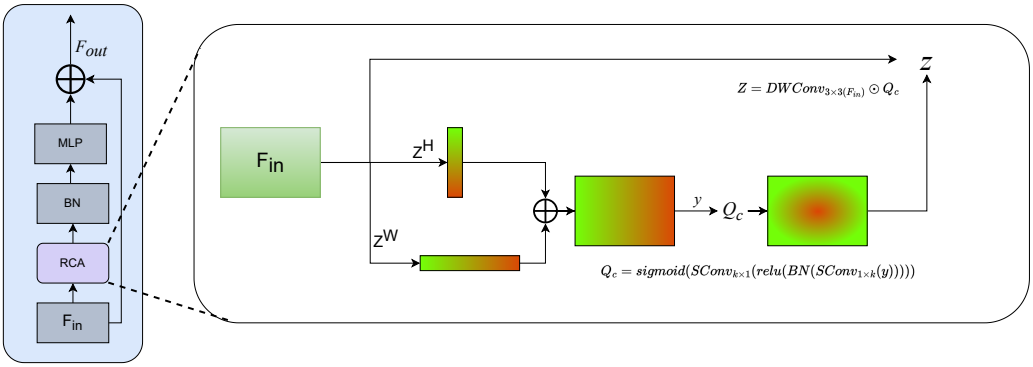


Figure 6. Components of the Rectangular Self-Calibration Module.

The modeling information obtained is calibrated through a self-calibration function to derive an attentional feature map that provides a more accurate representation of the foreground information. The self-calibration function incorporates strip convolutions in both the horizontal and vertical dimensions. At the outset, horizontal strip convolution is deployed to calibrate the shape in the horizontal plane. Subsequently, a BN layer is employed for normalization, and a ReLU activation function is introduced for non-linearity. Subsequently, vertical strip convolution is employed to calibrate the shape in the vertical dimension, with the application of the sigmoid function to introduce non-linearity. By decoupling the convolutions in these two dimensions, the number of parameters is reduced while enabling the model to adapt to the uncertain shape characteristics of smoke. The self-calibration function can be mathematically represented as follows:

$$Q_c = sigmoid(SConv_{k\times1}(relu(BN(SConv_{1\times k}(y))))) \tag{5}$$

where y represents the input of the self-calibration function, $SConv_{n\times m}$ represents the strip convolution, $sigmoid(x)$ represents the Sigmoid function, and $relu(x)$ represents the ReLU function.

Finally, a fusion function is utilized to integrate the original input with the output processed by the self-calibration function, thereby enhancing the weight of foreground features in the image. Specifically, 3×3 DepthWise Convolution (DWConv) is first employed to extract features from the original image. Then, it leverages the Hadamard product to compute the output of self-calibrated features, thereby weighting the foreground features. The fusion function can be mathematically represented as follows:

$$Z = DWConv_{3\times3(F_{in})} \odot Q_c \tag{6}$$

where $DWConv_{3\times3}$ represents DWConv with a 3×3 kernel, F_{in} represents the input of the network, the final variable signifies the Hadamard product.

To augment the representation of features, we incorporate a Batch BN layer and an MLP layer subsequent to the RCA and utilize a residual structure to form the RCM:

$$RCM(F_{in}) = MLP(BN(DWConv_{3\times3}(F_{in}) \odot Q_c)) + F_{in} \tag{7}$$

3.5.2. Multi-Scale Feature Foreground Enhancement

The multi-scale feature foreground Enhancement module centers around pyramid context extraction, as illustrated in Figure 7. It is responsible for extracting and fusing feature map information at different scales. Then, an enhanced representation of the foreground is obtained through RCM. As previously stated, the set of input features includes $\{S3, S4, F5\}$. Subsequently, $S3$, $S4$, and $F5$ are downsampled to the size of $\frac{H}{64} \times \frac{W}{64}$ through the application of average pooling and are then concatenated together to generate the pyramid feature $F6$. $F6$ is then fed into three repetitive RCMs for pyramid feature interaction to extract scale-aware semantic features. The following equation can describe this process:

$$S3', S4', F5' = Split(RCM(Concat(Avg(S3, 8), Avg(S4, 4), Avg(F5, 2)))) \tag{8}$$

where $Avg(F, x)$ represents average pooling with downsampling x times on feature map F.$S3'$. $S4'$ and $F5'$ are new feature maps with multi-scale perceptual information.

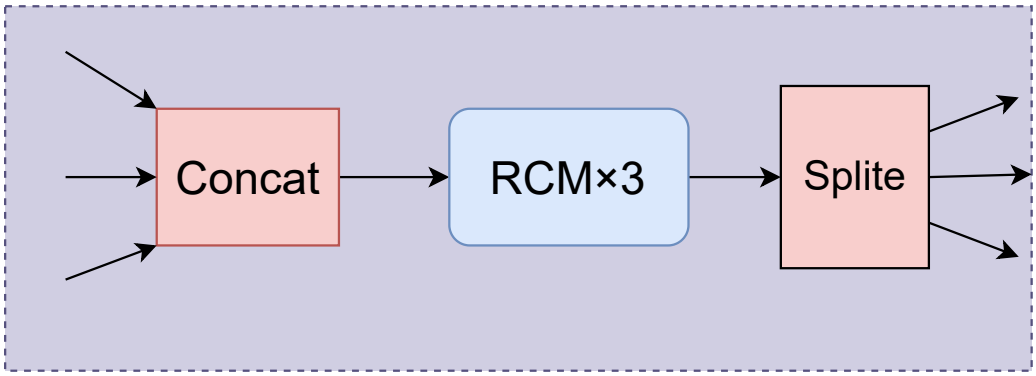


Figure 7. Details of pyramid context extraction.

3.5.3. Foreground Feature Fusion Module

In the Foreground Feature Fusion Module, feature maps of different scales are independently processed through RCMs and fused with multi-scale feature foreground Enhancement outputs. The feature fusion process begins with the $F5$ feature map. After processing through RCM, it undergoes interpolated multiplication with $F5'$ to generate $P5$. Subsequently, the $S4$ feature map performs interpolated addition with $P5$, followed by interpolated multiplication with $S4'$ to achieve feature fusion across different scales, resulting in $P4$. Similarly, after RCM processing, the $S3$ feature map undergoes interpolated addition with $P4$, followed by interpolated multiplication with $S3'$ to produce $P3$. Through these operations, we obtain feature maps $P3$, $P4$, and $P5$, which have undergone foreground focusing through dual pathways and further fusion. This process can be mathematically formulated as follows:

$$P5 = RCM(F5) \odot F5' \tag{9}$$

$$P4 = RCM(S4) \oplus P5 \odot S4' \tag{10}$$

$$P3 = RCM(S3) \oplus S4 \odot S3' \tag{11}$$

where $RCM(x)$ represents the processing of the input x through the RCM, $\oplus$ denotes interpolation addition, and $\odot$ denotes interpolation multiplication. Interpolation addition performs matrix addition on two elements after upsampling the feature map of a smaller

size, while interpolation multiplication performs matrix multiplication on two elements after upsampling the feature map of a smaller size.

Finally, $P3$, $P4$, and $P5$ are flattened from top to bottom and concatenated into a feature vector, which serves as the input for IoU-aware query selection. Specifically, after applying a single convolution operation to $P3$, $C1$ is obtained. $C1$ is then downsampled and concatenated with $P4$ along the channel dimension. This concatenated result is processed through a RepC3 module followed by a standard convolution operation to yield $C2$. Similarly, $C2$ is downsampled and concatenated with P5 along the channel dimension, and the concatenated result is further processed through a RepC3 module and a convolution operation to produce $C3$. Finally, $C1$, $C2$, and $C3$ are flattened into vectors and concatenated together to form the final result. The detailed process is as follows:

$$C1 = Conv(P3) \tag{12}$$

$$C2 = Conv(RepC3(Concat(DownSample(C1), P4))) \tag{13}$$

$$C3 = Conv(RepC3(Concat(DownSample(C2), P5))) \tag{14}$$

$$Out = Concat(Flatten(C1), Flatten(C2), Flatten(C3)) \tag{15}$$

where $Conv(x)$ represents the convolution operation for input x. $RepC3(x)$ represents input x through the RepC3 module. It is a convolutional block that uses RepConv.

4. Experiments and Results

4.1. Experimental Equipment and Hyper-Parameter Settings

In terms of the hardware employed in this study, the CPU model utilized is the Intel® Xeon® CPU E5-2680 v4, the memory size is 30G, and the GPU model is the RTX 3090, equipped with 24G video memory. Regarding software, the Python version employed in this study was 3.8, the CUDA version was 12.4, and the Pytorch version was 2.4. In the context of deep learning, the configuration of hyper-parameters is important. During the training phase, images of dimensions $3 \times 640 \times 640$ were employed as inputs, with a batch size of 4. Following 200 epochs of training, each epoch of training required approximately 120 s. The initial learning rate (LR0) was set to 0.0001, and the final learning rate (LR1) was set to 1.0. The optimizer used for training was AdamW, the momentum size was set to 0.9, and Weight Decay was set to 0.0001. These settings were maintained by the RT-DETR default settings.

4.2. Dataset and Evaluation Metrics

4.2.1. Smoke Detection Dataset

This study is dedicated to enhancing the accuracy of smoke detection by meticulously crafting a dataset that comprises diverse and high-quality images. This dataset integrates smoke images sourced from various origins, including the authoritative dataset released by Feiniu Yuan and his team at the State Key Laboratory of Fire Science at the University of Science and Technology of China, smoke images from surveillance perspectives provided by the Roboflow platform, and smoke images extensively collected from the internet by us and rigorously annotated. During the construction of the dataset, we observed that public datasets typically comprise frames extracted from multiple video segments, resulting in a significant number of temporally consecutive and highly similar frame images. This could potentially lead to issues of uneven sample distribution and data redundancy. We adopted a scientific proportional selection strategy to address this, effectively mitigating these issues. We also specifically incorporated non-smoke sample images, some of which contain objects resembling smoke, with the aim of enhancing the model's anti-interference capability and reducing the false detection rate. Through a series of meticulous data processing and optimization efforts, we successfully constructed a high-quality, diverse, and comprehensively covered smoke dataset. Figure 8 illustrates a selection of images from the dataset, which encompass a diverse range of perspectives and scenarios of smoke,

thereby augmenting the model's capacity to discern smoke in various environments. This dataset contains 4874 smoke images, divided into a training set of 3411 images, a validation set of 731 images, and a test set of 732 images. It provides a solid foundation for the subsequent training of smoke detection models. Figure 9 illustrates a scatter plot of the location and size distribution of the actual labels in the dataset. This figure demonstrates that the location of the smoke in the dataset is uniformly distributed and that the majority of smoke is of a small or medium size. This allows for timely detection of smoke, which can then be used to issue early warnings of fires.

Figure 8. Presentation of smoke images within the dataset.

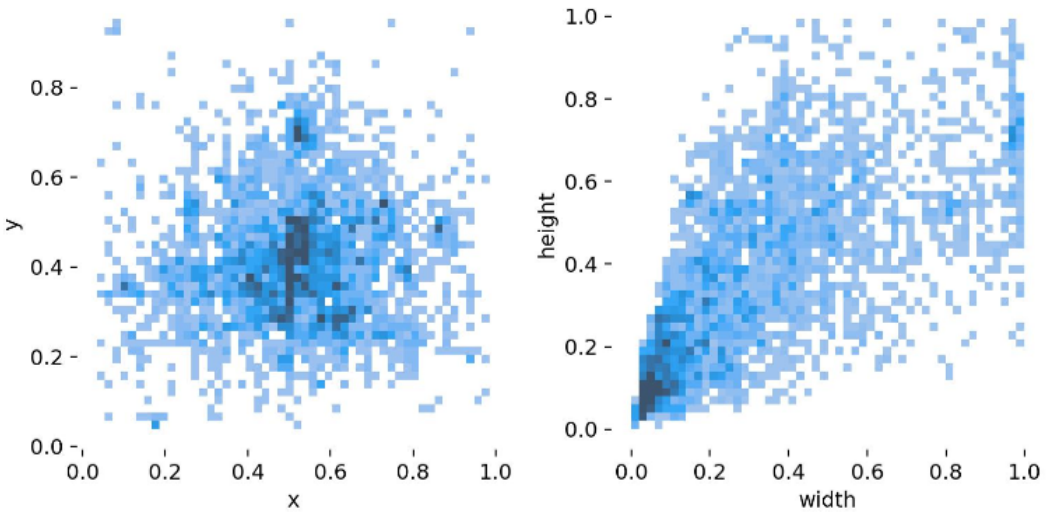


Figure 9. Scatterplot of location and size distribution of actual labels.

4.2.2. Evaluation Metrics

To evaluate the performance of the model in smoke detection, the object detection evaluation metrics used in this paper are Precision, Recall, and Mean Average Precision (mAP).

Precision reflects the proportion of instances predicted by the model to be positive samples that are truly positive, and the calculation formula is expressed as

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

where true positive (TP) denotes the number of instances correctly predicted to be smoke and false positive (FP) denotes the number of instances incorrectly predicted to be smoke.

Recall measures the proportion of all true positive samples that are correctly predicted to be positive by the model, and its calculation formula is expressed as

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

where false negative (FN) represents the number of instances that are not predicted to be smoke by the model but are actually smoke.

Mean Average Precision is an evaluation metric in the field of object detection that combines Precision and Recall to comprehensively assess the performance of a model. It calculates the Average Precision at various Recall levels to provide a comprehensive evaluation. The calculation formula for mAP can be expressed as follows:

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{18}$$

$$AP = \int_0^1 P(R)dR \tag{19}$$

There is only one category in smoke detection, so mAP is equivalent to Average Precision (AP). Depending on different Intersection over Union (IoU) thresholds, common mAP metrics include mAP50 and mAP95. IoU is a metric that quantifies the degree of overlap between a predicted bounding box and a ground truth bounding box, with values ranging from 0 to 1. In the context of smoke detection, mAP50 refers to the mAP value calculated when the IoU threshold is set to 0.5. Conversely, mAP95 represents the mAP value computed over a range of IoU thresholds from 0.5 to 0.95.

4.3. Comparative Experiments

4.3.1. Evaluation in Comparison with a Number of Advanced Object Detection Algorithms

To comprehensively evaluate the performance of Smoke-DETR in the task of smoke detection, we compared it with various advanced object detection models. We selected Faster-Rcnn [20], Yolov8 [37], Yolov9 [38], Yolov11 [37], Rtmdet [39], Detr with Improved deNoising AnchOr Boxes (DINO) [40], and the baseline model RT-DETR for experimentation alongside Smoke-DETR. The detailed experimental results are presented in Table 1. The best results are highlighted in bold.

Table 1. Comparison of experimental results of multiple advanced object detection models with Smoke-DETR.

Model	Precision (%)	Recall (%)	mAP50 (%)	mAP95 (%)	Parameters	Flops
Faster-Rcnn	0.798	0.726	0.772	0.476	40 M	207 G
Yolov8m	0.813	0.784	0.826	0.527	25.8 M	78.7 G
Yolov9m	0.823	0.79	0.836	0.516	16.6 M	60.0 G
Yolov11m	0.819	0.765	0.829	0.533	20.1 M	68.0 G
Rtmdet	0.802	0.743	0.817	0.500	8.89 M	14.8 G
DINO	**0.876**	0.776	0.841	0.532	47 M	279 G
RT-DETR	0.832	0.764	0.824	0.517	19.9 M	56.9 G
Smoke-DETR	0.868	**0.8**	**0.862**	**0.539**	16.4 M	43.3 G

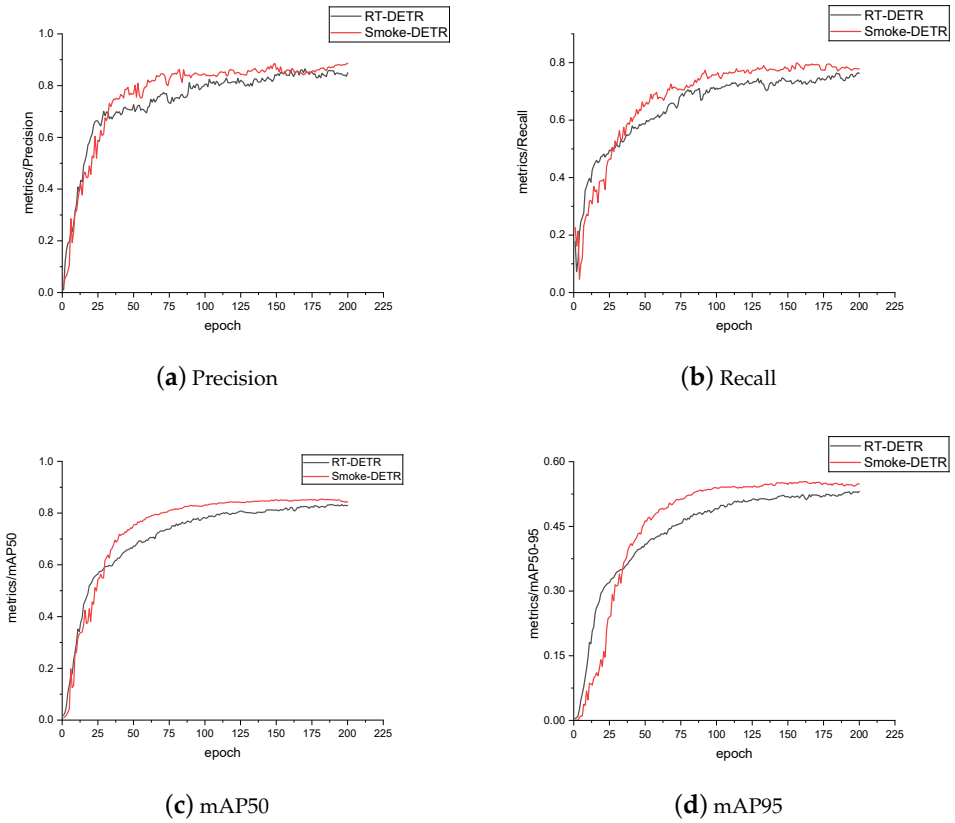
We examined Faster R-CNN as a paradigm of early object detection technology, but, unfortunately, its accuracy performance in smoke detection tasks is less than satisfactory. To further delve into the analysis, we also compared advanced object detection models that have emerged in the past three years. Among them, YOLOv8, YOLOv9, and YOLOv11, as the latest models in the YOLO series, rank among the top in multiple performance indicators, especially with YOLOv9 achieving suboptimal results in Recall, and YOLOv11 ranking second in mAP95. Rtmdet, an optimized version based on You Only Look Once version X(YOLOX) [41], is characterized by its low parameter count and computational complexity. However, it does not demonstrate significant advantages in the four key evaluation indicators (Recall, Precision, mAP50, and mAP95). On the other hand, DINO, as a leader in the DETR series, holds the top spot in Precision and ranks second in mAP50. However, its high parameter and computational complexity cannot meet real-time demands. RT-DETR finds a good balance between performance and efficiency, with moderate performance in its four indicators and lower parameters and computational costs, providing a solid foundation for subsequent model improvements. Based on RT-DETR, we propose Smoke-DETR, which achieves optimal levels in the three indicators of Recall, mAP50, and mAP95. Specifically, our model outperforms the second-ranked model by one percentage point in Recall, leads by 2.1 percentage points in mAP50, and surpasses the second-ranked model by 0.6 percentage points in mAP95. These data demonstrate the applicability of Smoke-DETR in smoke detection among multiple excellent object detection models.

4.3.2. Comparison of the Baseline Model

Figure 10 illustrates the trends in Precision, Recall, and mAP of the validation set across various training epochs. In the initial stages of training, RT-DETR demonstrated a certain leading edge; however, after the 25th epoch, RT-DETR was gradually surpassed by Smoke-DETR, and this trend continued until the end of training. By further observing the changes in mAP, it can be found that Smoke-DETR had already converged around the 100th epoch, while RT-DETR showed signs of convergence only around the 125th epoch. This indicates that not only did Smoke-DETR outperform RT-DETR in the final performance, but it also had a significant advantage in model convergence speed. Smoke-DETR was able to achieve higher Precision, Recall, and mAP in fewer training epochs, suggesting that it is more efficient at learning key features from the data.

Figure 11 demonstrates the comparison between the final results of RT-DETR and Smoke-DETR. Compared with the baseline model, this model improves Precision by 3.6 percentage points, Recall by 3.6 percentage points, mAP50 by 3.8 percentage points, and mAP95 obtains an improvement of 2.2 percentage points. Figure 12 demonstrates a comparison of the number of parameters and the computation between the two models, and it is obvious that the number of parameters and the computation of Smoke-DETR have been significantly decreased. In other words, Smoke-DETR uses fewer resources to achieve better results.

Figure 13 presents a comparison of the performance between RT-DETR and Smoke-DETR in the task of smoke detection. A series of representative images were carefully selected to clearly highlight the outstanding performance of Smoke-DETR. In complex scenes containing various scales and multiple smoke targets (as shown in the first and third rows), RT-DETR exhibited significant missed detections. In contrast, Smoke-DETR was able to accurately identify all smoke targets, with a particularly prominent advantage in detecting small-sized smoke targets. When faced with the more challenging task of thin fog target detection (as shown in the second and sixth rows), RT-DETR's detection results deviated significantly from the actual situation, with unsatisfactory accuracy. Smoke-DETR, on the other hand, demonstrated higher detection Precision, with results that were more consistent with reality and a significant improvement in accuracy. Additionally, in dealing with some common smoke images (as shown in the fourth and fifth rows), Smoke-DETR also showed superior recognition accuracy to RT-DETR, further confirming its exceptional performance and broad application potential in the field of smoke detection.



(**a**) Precision (**b**) Recall

(**c**) mAP50 (**d**) mAP95

Figure 10. The results of valid sets for RT-DETR and Smoke-DETR during the training process.

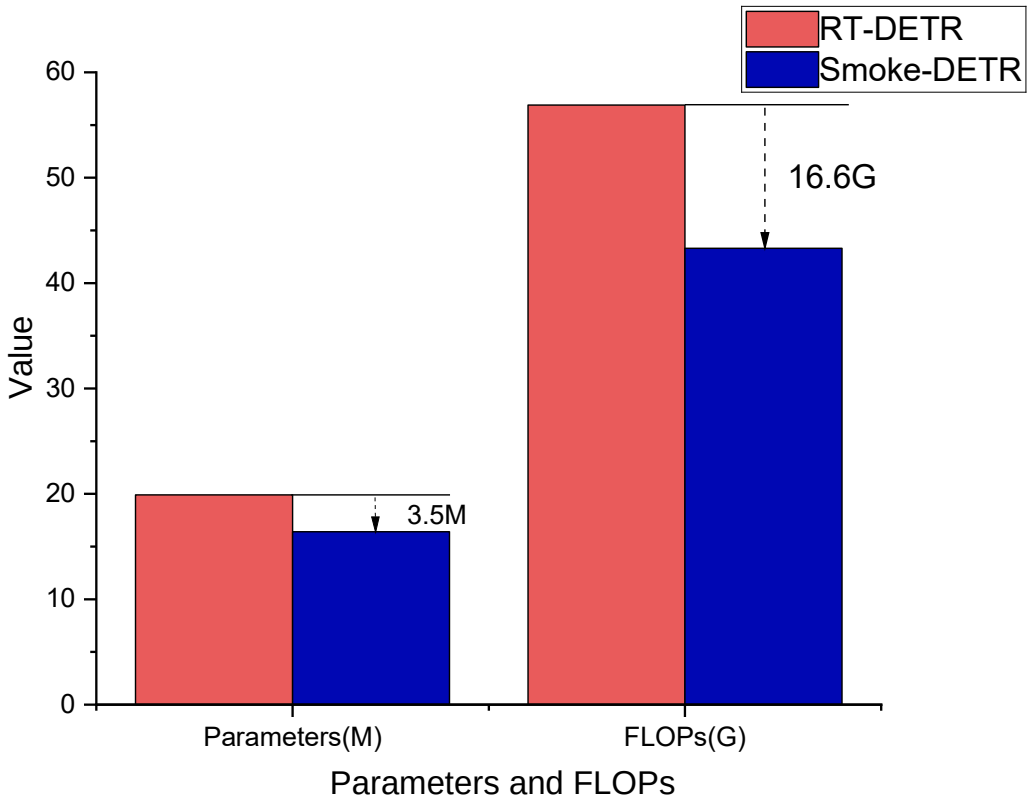


Figure 11. Comparison of parameters and FLOPs between RT-DETR and Smoke-DETR.

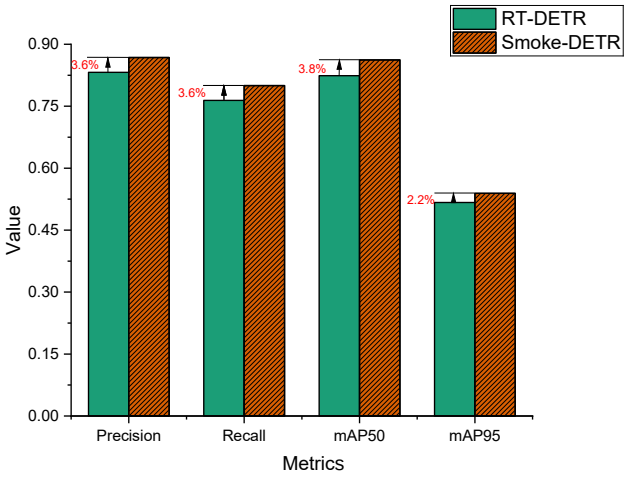
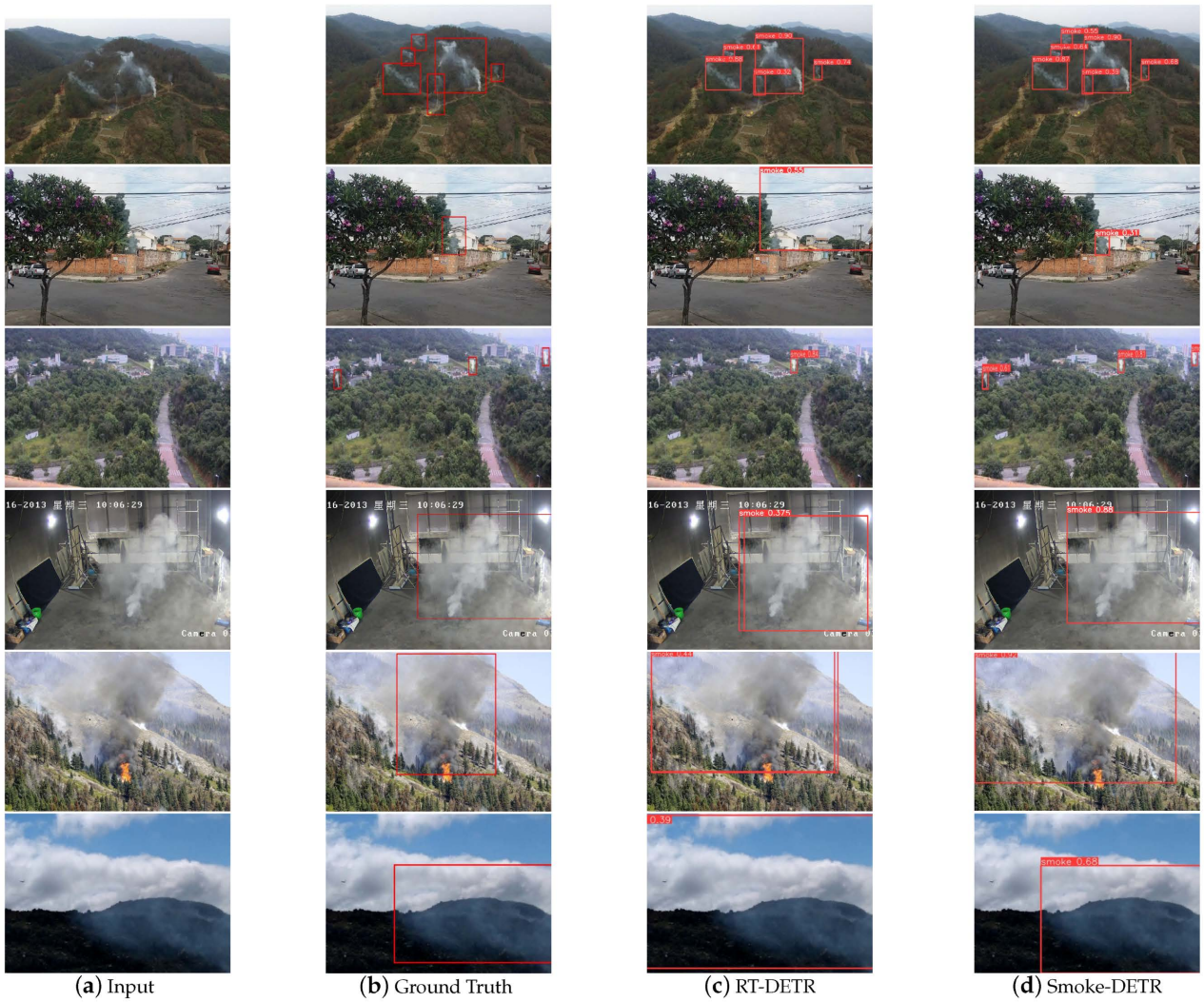


Figure 12. Comparison of RT-DETR and Smoke-DETR in terms of four metrics: Precision, Recall, mAP50, and mAP95.



(**a**) Input (**b**) Ground Truth (**c**) RT-DETR (**d**) Smoke-DETR

Figure 13. The detection results using different methods. (**a**) The input image to be detected. (**b**) The manually labeled ground truth. (**c**) The detection result of RT-DETR. (**d**) The detection result of Smoke-DETR.

4.4. Ablation Experiment

In order to investigate in depth the impact of different improvement measures on the performance of the RT-DETR model and to validate the effectiveness of our proposed methods, we selected the RT-DETRr18 model as the baseline model and applied different improvement strategies individually to determine the specific improvement effects of each modification on model performance. To ensure fairness of comparison, all experiments were performed under identical hyper-parameter configurations, and the models were trained from scratch. The experimental results are shown in Table 2.

Firstly, we introduced the Multi-Scale Feature Fusion Pyramid Network (MFFPN) to replace the feature fusion module of the baseline model. This substitution led to significant improvements across multiple performance metrics: Precision increased by 3.1 percentage points, Recall rose by 3.4 percentage points, mAP50 saw a 2.5 percentage point boost, and mAP95 improved by 1.3 percentage points. These enhancements demonstrate the effectiveness of the multi-scale effective fusion and foreground-focused strategies in the context of smoke detection. Moreover, by incorporating stripe convolutions in both horizontal and vertical directions within the feature fusion process to decouple the convolutions, the computational load is reduced efficiently. Specifically, the computational complexity decreased from 56.9 G to 48.2 G, representing a notable 15.2% reduction.

Subsequently, we replaced the ResNetBlock in the baseline model with the ECPConvBlock without EMA. This led to a reduction in model parameters from 19.8 million to 16.7 million. Additionally, the Precision increased by 1.9 percentage points, the Recall by 1.2 percentage points, mAP50 by 1.9 percentage points, and mAP95 by 0.4 percentage points. These results indicate that ECPConv not only reduces the number of model parameters but also achieves an increase in accuracy.

Next, we introduced EMA into the backbone network, resulting in improvements of 4.2 percentage points in Precision, 2.1 percentage points in Recall, 2.3 percentage points in mAP50, and 0.9 percentage points in mAP95, respectively. This validates the enhancement effect of EMA on the model's feature extraction capabilities.

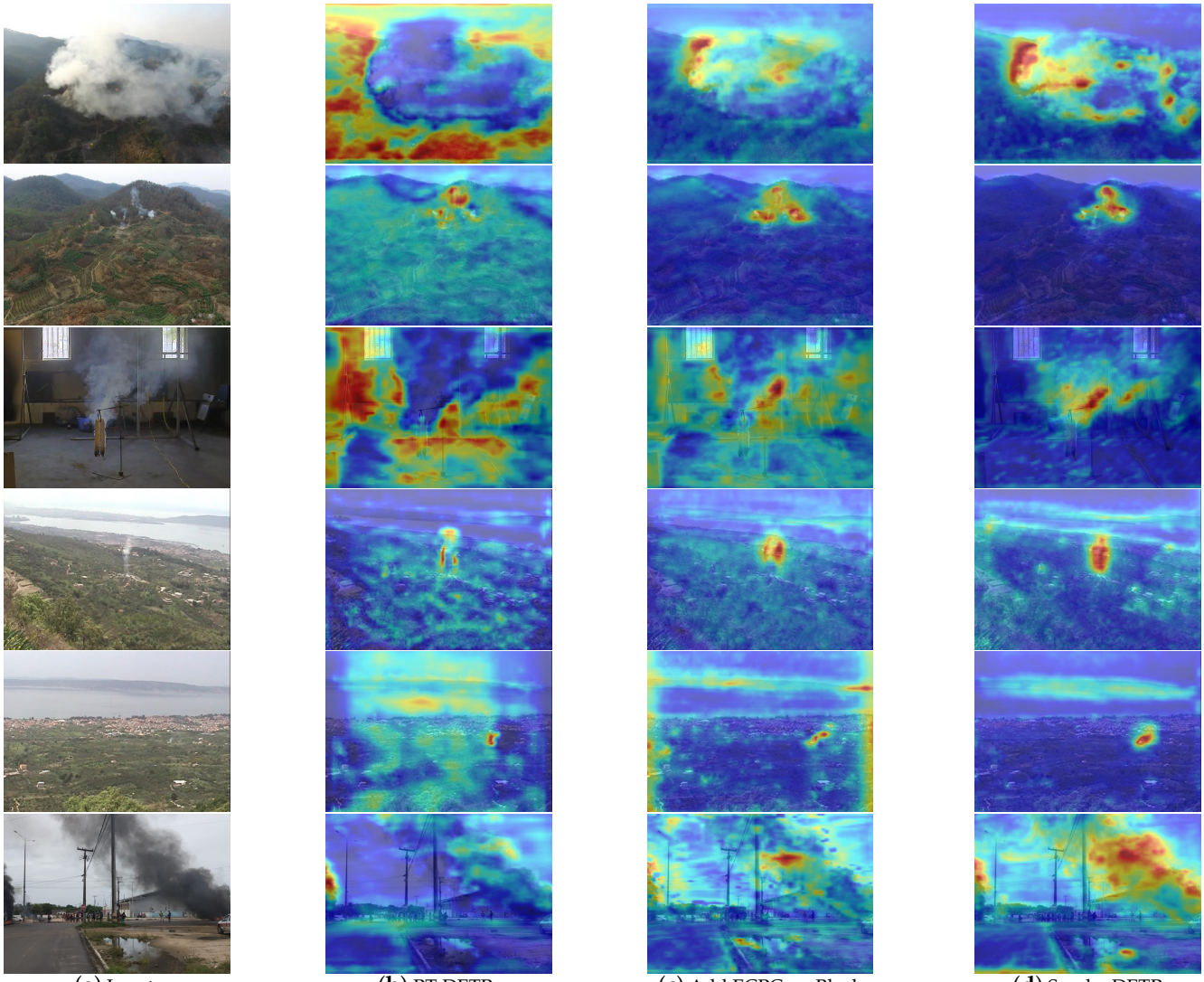
Finally, we comprehensively applied these three improvement measures to the model, achieving the best results. Specifically, Precision and Recall increased by 3.6 percentage points, mAP50 increased by 3.8 percentage points, and mAP95 increased by 2.2 percentage points. Meanwhile, the number of parameters was reduced by 17 percentage points, and the computational load decreased by 23.9 percentage points. These results demonstrate that our improvement strategies effectively enhance the performance of the smoke detection model.

Table 2. Effect of different improvements on model performance.

Method	Precision (%)	Recall (%)	mAP50 (%)	mAP95 (%)	Parameters	Flops
Rtdetr-r18	0.832	0.764	0.824	0.517	19.9 M	56.9 G
Rtdetr-r18 + MFFPN	0.875	0.791	0.847	0.527	19.2 M	48.2 G
Rtdetr-r18 + ECPConv	0.842	0.793	0.840	0.522	16.8 M	49.5 G
Rtdetr-r18 + EMA	0.851	0.791	0.842	0.525	20.1 M	58.1 G
Rtdetr-r18 + (MFFPN + ECPConv + EMA)	0.868	0.800	0.862	0.539	16.3 M	43.3 G

We utilized Gradient-weighted Class Activation Mapping (Grad-CAM) technology to generate heatmaps, which visualize the regions of interest that the model focuses on during the smoke detection task. The deeper the color in the heatmap, the more significant the contribution of that region to the model's predictions, thus clearly revealing the model's "visual focus". By observing the heatmaps, we can intuitively see the distribution of the model's attention points when processing different images. Figure 14 compares the heatmap results of different methods in the smoke detection task with test images, including smoke images with low contrast against the background (first row), small smoke images (second, fourth, and fifth rows), thin smoke images (third row), and multi-scale smoke images (sixth row).

The original RT-DETR model struggles to focus on the target when dealing with smoke that is poorly distinguished from the background. However, the model can roughly attend to the smoke region after improving the backbone network. With the further introduction of MFFPN, the color of the smoke region intensifies in the heatmap, indicating a significant increase in the model's attention to the smoke target.



(**a**) Input (**b**) RT-DETR (**c**) Add ECPConvBlock (**d**) Smoke-DETR

Figure 14. Heat map results under different methods. A darker shade of red indicates a higher probability ascribed by the model to the presence of smoke in that particular section. (**a**) Input image. (**b**) The heat map of RT-DETR model). (**c**) The heat map of backbone with ECPConvBlock. (**d**) The heat map of further introducing MFFPN.

When facing small smoke targets, RT-DETR's attention is widely distributed, making it difficult to lock onto the target. After improving the backbone network, the model focuses on the target region. MFFPN further enhances the color depth of the smoke region, improving the detection capability for small smoke targets. While the improved backbone network can focus on some regions for thin smoke targets, its effect is limited. In contrast, MFFPN directly distinguishes between the background and the smoke target, enabling the model to accurately focus on the smoke target and demonstrating its potential in detecting complex smoke morphologies.

When dealing with multi-scale smoke, RT-DETR can locate the smoke region, but the color of the target region is relatively light, resulting in weak detection performance.

Improving the backbone network increases the attention to the target region. However, the attention to the background has not decreased. After adding MFFPN, the attention to non-target regions decreases, and the attention to target regions deepens, significantly improving the detection accuracy of multi-scale smoke.

In summary, the heatmaps intuitively demonstrate the impact of different improvement methods on the detection characteristics of the model. The EMA exhibits a certain degree of adaptability when dealing with multi-scale smoke. At the same time, MFFPN further enhances the model's detection accuracy by focusing on the foreground and increasing the distinction between smoke and the background.

4.4.1. Experiments in the Feature Fusion Section

We integrated a Bi-directional Feature Pyramid Network (BiFPN) [42] and a Multi-branch Assisted Feature Pyramid Network (MAFPN) from MAF-YOLO [43] into the RT-DETR model for comparison with MFFPN, as evidenced by the results presented in Table 3. The experimental outcomes indicate that our MFFPN not only attains superior performance among various competitive feature fusion architectures but also diminishes the model's parameter count.

Table 3. Experimental results of different feature fusion methods.

Method	Precision (%)	Recall (%)	mAP50 (%)	mAP95 (%)	Parameters	Flops
PANET (baseline)	0.832	0.764	0.824	0.517	19.9 M	56.9 G
MFFPN (ours)	0.875	0.791	0.847	0.527	19.2 M	48.2 G
MAFPN	0.845	0.783	0.835	0.513	22.9 M	56.3 G
BIFPN	0.862	0.805	0.843	0.518	20.3 M	64.3 G

4.4.2. Experiments on the Backbone

To delve into the effectiveness of integrating channel selection strategies within ECPConv and the enhancement of the main backbone network performance by our proposed ECPConvBlock, we conducted experiments on various backbone networks. We tested them by replacing ResNetBlock with FasterBlock and then evaluated the performance by substituting PConv with ECPConv within FasterBlock. Finally, we constructed ECPConvBlock with EMA and verified it through experiments. As shown in Table 4, the results indicate that ECPConv improves the feature extraction capability of FasterBlock, and the combination of EMA with ECPConv further enhances the model accuracy.

Table 4. Experimental results of different backbones.

Method	Precision (%)	Recall (%)	mAP50 (%)	mAP95 (%)	Parameters	Flops
Rtdetr-r18 (baseline)	0.832	0.764	0.824	0.517	19.9	56.9
PConv	0.847	0.769	0.832	0.520	16.8	49.5
ECPConv	0.851	0.776	0.843	0.521	16.8	49.9
ECPConv + EMA	0.851	0.791	0.852	0.535	17.0	51.4

4.4.3. Experiments with Different Iou Loss Function

Many object detection models are focused on enhancing the loss function to improve detection accuracy. This approach offers a non-destructive improvement method, particularly for the Iou loss. In order to ascertain which loss function is most appropriate for the smoke detection task, experiments were conducted on a range of loss functions, incorporating all the aforementioned improvements. These included Minimum Point Distance-based IoU (MPDIoU) [44], Focal and efficient IOU (Focal-EIOU) [45], Shape-IoU [46], and Normalized Gaussian Wasserstein Distance Loss (NWDLoss) [47]. The results of these experiments are presented in Table 5. The results demonstrate that the GIou is the most suitable loss function.

Table 5. Experimental results for different Iou loss functions.

Method	Precision (%)	Recall (%)	mAP50 (%)	mAP95 (%)
Smoke-DETR (GIou)	0.868	0.800	0.862	0.539
Smoke-DETR (MDPIou)	0.875	0.781	0.850	0.534
Smoke-DETR (EIou)	0.849	0.801	0.841	0.536
Smoke-DETR (Shape-Iou)	0.851	0.776	0.843	0.521
Smoke-DETR (NWDLoss)	0.888	0.777	0.846	0.511

5. Conclusions

Based on the RT-DETR model, this paper presents Smoke-DETR, a novel object detection model specifically designed for smoke detection in outdoor early fire warning systems. The model incorporates various improvements targeting the unique characteristics of smoke. Considering the computational limitations of outdoor edge devices, we introduced improvements using Partial Convolution concepts without significantly increasing model parameters and computational complexity. We proposed the ECPConv for the backbone network using a channel selection strategy. To address the challenges of irregular smoke shapes and subtle features, we incorporated a novel EMA to enhance the feature extraction capabilities of the backbone network. While these backbone improvements significantly increased the model's Precision, Recall, and mAP, detection errors still occurred when background and foreground similarities were high. To address this problem, we developed the MFFPN based on RCM to enhance the model's foreground focus capabilities. Extensive experiments demonstrated the superior performance of Smoke-DETR in smoke detection. Firstly, Smoke-DETR achieved the best results on multiple metrics in comparative experiments with advanced models. Compared to the baseline RT-DETR model, Smoke-DETR showed improvements of 3.6, 3.6, 3.8, and 2.2 percentage points in Precision, Recall, mAP50, and mAP95 respectively. It also reduced parameters by 17% and computational complexity by 23.9%. Detection result visualizations demonstrated Smoke-DETR's superior smoke detection accuracy compared to that of RT-DETR. Secondly, ablation experiments were designed to explore the performance improvements of the model progressively after incorporating ECPConv, EMA, and MFFPN individually, as well as when integrating all three methods. Meanwhile, the heatmaps also demonstrated the model's enhanced ability to focus on smoke with the sequential addition of different improvements. Finally, some comparative experiments were conducted separately on the backbone network, feature fusion network, and loss function, and the methods we adopted all exhibited higher accuracy. The comprehensive experimental results, including comparison, ablation, and module-specific experiments supported by detection visualizations and heatmaps, demonstrate the exceptional performance of Smoke-DETR in smoke detection.

Several directions for future research have been identified. First, the dataset size can be expanded by continuously collecting early-stage fire smoke images to improve both quality and quantity. Second, while the current model maintains real-time detection capabilities with reduced parameters and computational costs, there is potential for further network optimization. Future research will explore pruning and distillation methods for Smoke-DETR to further reduce model parameters and facilitate deployment on edge devices such as surveillance equipment and drones. Finally, we plan to investigate the relationship between flames and smoke in early fire stages to develop more precise early fire detection algorithms, contributing to fire prevention and disaster reduction efforts.

Author Contributions: Conceptualization, B.S. and X.C.; methodology, B.S.; software, X.C.; validation, B.S. and X.C.; formal analysis, B.S.; investigation, B.S. and X.C.; resources, B.S.; data curation, B.S. and X.C.; writing—original draft preparation, B.S. and X.C.; writing—review and editing, B.S. and X.C.; visualization, X.C.; supervision, B.S.; project administration, B.S.; funding acquisition, B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by Natural Science Foundation of China Grants (61972456 and 61173032) and by the Tianjin Natural Science Foundation (20JCYBJC00140).

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: The datasets used during the current study are available from the corresponding author on reasonable request.

Acknowledgments: We would like to thank the School of Computer Science and Technology, Tiangong University, for supporting our work.

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

Appendix A

Algorithm A1 The Pseudocode of Enhanced Channel-wise Partial Convolution

Input: $X = \{x_1, x_2, x_3, \ldots, x_c\}$
Output: $X' = \{x_1, x_2', x_3, \ldots, x_k', \ldots, x_c'\}$

Function ECPConv(X)
Y = []
for x **in** X **do**
    $i = 0$
    $y_i = Avg_c(x)$
    $Y.add(i, y_i)$
end for
  $Sigmoid(FC(ReLu(FC(Y))))$
  sort Y by y_i
  Z = []
for k **in** rang$(0, \frac{c}{4})$ **do**
    Z.add(X[Y[k]])
    $RepConv(Z)$
    $X[Y[k]] = Z[k]$
end for
return X
end Function

References

1. Yuan, F.; Shi, J.; Xia, X.; Zhang, L.; Li, S. Encoding pairwise Hamming distances of Local Binary Patterns for visual smoke recognition. *Comput. Vis. Image Underst.* **2019**, *178*, 43–53. https://doi.org/10.1016/j.cviu.2018.10.008.
2. Yuan, F.; Shi, J.; Xia, X.; Fang, Y.; Fang, Z.; Mei, T. High-order local ternary patterns with locality preserving projection for smoke detection and image classification. *Inf. Sci.* **2016**, *372*, 225–240. https://doi.org/10.1016/j.ins.2016.08.040.
3. Natural Resources Canada National Wildland Fire Situation Report. 2023. Available online: https://cwfis.cfs.nrcan.gc.ca/report (accessed on 5 March 2023).
4. Barbero, R.; Abatzoglou, J.T.; Larkin, N.K.; Kolden, C.A.; Stocks, B. Climate change presents increased potential for very large fires in the contiguous United States. *Int. J. Wildland Fire* **2015**, *24*, 892–899. https://doi.org/10.1071/WF15083.
5. U.S. Fire Administration. Fire Deaths, Fire Death Rates, and Risk of Dying in a Fire. 2020. Available online: https://www.usfa.fema.gov/statistics/deaths-injuries/states.html (accessed on 1 July 2024).
6. Chen, S.; Cao, Y.; Feng, X.; Lu, X. Global2Salient: Self-adaptive feature aggregation for remote sensing smoke detection. *Neurocomputing* **2021**, *466*, 202–220. https://doi.org/10.1016/j.neucom.2021.09.026.
7. Asiri, N.; Bchir, O.; Ismail, M.M.B.; Zakariah, M.; Alotaibi, Y.A. Image-based smoke detection using feature mapping and discrimination. *Soft Comput.* **2021**, *25*, 3665–3674. https://doi.org/10.1007/s00500-020-05396-4.
8. Carletti, V.; Greco, A.; Saggese, A.; Vento, B. A smart visual sensor for smoke detection based on deep neural networks. *Sensors* **2024**, *24*, 4519. https://doi.org/10.3390/s24144519.
9. Saydirasulovich, S.N.; Mukhiddinov, M.; Djuraev, O.; Abdusalomov, A.; Cho, Y.I. An improved wildfire smoke detection based on YOLOv8 and UAV images. *Sensors* **2023**, *23*, 8374. https://doi.org/10.3390/s23208374.
10. Chen, J.; Kao, S.H.; He, H.; Zhuo, W.; Wen, S.; Lee, C.H.; Chan, S.H.G. Run, don't walk: Chasing higher FLOPS for faster neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 July 2023; pp. 12021–12031.

11. Maruta, H.; Nakamura, A.; Kurokawa, F. A new approach for smoke detection with texture analysis and support vector machine. In Proceedings of the 2010 IEEE International Symposium on Industrial Electronics, Bari, Italy, 4–7 July 2010; pp. 1550–1555. https://doi.org/10.1109/ISIE.2010.5636301.
12. Tian, H.; Li, W.; Ogunbona, P.O.; Wang, L. Detection and Separation of Smoke from Single Image Frames. *IEEE Trans. Image Process.* **2018**, *27*, 1164–1177. https://doi.org/https://doi.org/10.1109/TIP.2017.2771499.
13. Jia, Y.; Yuan, J.; Wang, J.; Fang, J.; Zhang, Q.; Zhang, Y. A saliency-based method for early smoke detection in video sequences. *Fire Technol.* **2016**, *52*, 1271–1292. https://doi.org/10.1007/s10694-014-0453-y.
14. Chunyu, Y.; Jun, F.; Jinjun, W.; Yongming, Z. Video fire smoke detection using motion and color features. *Fire Technol.* **2010**, *46*, 651–663. https://doi.org/10.1007/s10694-009-0110-z.
15. Li, T.; Zhao, E.; Zhang, J.; Hu, C. Detection of Wildfire Smoke Images Based on a Densely Dilated Convolutional Network. *Electronics* **2019**, *8*, 1131. https://doi.org/10.3390/electronics8101131.
16. Wang, G.; Yuan, F.; Li, H.; Fang, Z. A pyramid Gaussian pooling based CNN and transformer hybrid network for smoke segmentation. *IET Image Process.* **2024**, *18*, 3206–3217. https://doi.org/10.1049/ipr2.13166.
17. 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.
18. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
19. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
20. 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.* **2017**, *39*, 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031.
21. Redmon, J. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.U.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
23. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.
24. Huang, J.; Zhou, J.; Yang, H.; Liu, Y.; Liu, H. A small-target forest fire smoke detection model based on deformable transformer for end-to-end object detection. *Forests* **2023**, *14*, 162. https://doi.org/10.3390/f14010162.
25. Liang, T.; Zeng, G. FSH-DETR: An Efficient End-to-End Fire Smoke and Human Detection Based on a Deformable DEtection TRansformer (DETR). *Sensors* **2024**, *24*, 4077. https://doi.org/10.3390/s24134077.
26. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs Beat YOLOs on Real-time Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–24 June 2024; pp. 16965–16974.
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part IV 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 630–645.
28. Xia, Z.; Pan, X.; Song, S.; Li, L.E.; Huang, G. Vision transformer with deformable attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4794–4803.
29. 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.
30. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
31. Chen, L.; Zhang, H.; Xiao, J.; Nie, L.; Shao, J.; Liu, W.; Chua, T.S. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5659–5667.
32. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient multi-scale attention module with cross-spatial learning. In Proceedings of the ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; IEEE: New York, NY, USA, 2023; pp. 1–5.
33. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the EEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
34. Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; Feng, J. Aˆ 2-Nets: Double Attention Networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2018; Volume 31.

35. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
36. Ni, Z.; Chen, X.; Zhai, Y.; Tang, Y.; Wang, Y. Context-Guided Spatial Feature Reconstruction for Efficient Semantic Segmentation. *arXiv* **2024**, arXiv:2405.06228.
37. Jocher, G.; Qiu, J.; Chaurasia, A. Ultralytics YOLO. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 10 January 2023).
38. Wang, C.Y.; Yeh, I.H.; Mark Liao, H.Y. Yolov9: Learning what you want to learn using programmable gradient information. In Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024; Springer: Berlin/Heidelberg, Germany, 2024; pp. 1–21.
39. Lyu, C.; Zhang, W.; Huang, H.; Zhou, Y.; Wang, Y.; Liu, Y.; Zhang, S.; Chen, K. Rtmdet: An empirical study of designing real-time object detectors. *arXiv* **2022**, arXiv:2212.07784.
40. Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.M.; Shum, H.Y. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv* **2022**, arXiv:2203.03605.
41. Ge, Z. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
42. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–16 June 2020; pp. 10781–10790.
43. Xue, Y.; Ju, Z.; Li, Y.; Zhang, W. MAF-YOLO: Multi-modal attention fusion based YOLO for pedestrian detection. *Infrared Phys. Technol.* **2021**, *118*, 103906. https://doi.org/10.1016/j.infrared.2021.103906.
44. Ma, S.; Xu, Y. Mpdiou: A loss for efficient and accurate bounding box regression. *arXiv* **2023**, arXiv:2307.07662.
45. Zhang, Y.F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* **2022**, *506*, 146–157. https://doi.org/10.1016/j.neucom.2022.07.042.
46. Zhang, H.; Zhang, S. Shape-iou: More accurate metric considering bounding box shape and scale. *arXiv* **2023**, arXiv:2312.17663.
47. Wang, J.; Xu, C.; Yang, W.; Yu, L. A normalized Gaussian Wasserstein distance for tiny object detection. *arXiv* **2021**, arXiv:2110.13389.