IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 August 2023, accepted 28 September 2023, date of publication 5 October 2023, date of current version 13 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3322143

RESEARCH ARTICLE

YOLO-SF: YOLO for Fire Segmentation Detection

XIANGHONG CAO, YIXUAN SU, XIN GENG, AND YONGDONG WANG
College of Building Environment Engineering, Zhengzhou University of Light Industry, Zhengzhou, Henan 450006, China

Corresponding author: Xin Geng (gengxin@email.zzuli.edu.cn)

This work was supported in part by the Science and Technology Research Project of the Henan Province, in part by the Research on Key Technologies for Fire Detection Based on Multi-Spectral Image Fusion under Grant 232102321021, and in part by the Research on Key Technologies of Three-Dimensional Spatial Temperature Field Reconstruction System Under High Temperature and Smoke Environment of Building Fire under Grant 222102220071.

ABSTRACT Owing to the problems of missed detection, false detection, and low accuracy of the current fire detection algorithm, a segmentation detection algorithm, YOLO-SF, is proposed. This algorithm combines instance segmentation technology with the YOLOv7-Tiny object detection algorithm to improve its accuracy. We gather images that include both fire and non-fire elements to create a fire segmentation dataset (FSD). The segmentation detection head of YOLOR is adopted to improve the accuracy of model segmentation and enhance its ability to express details. The MobileViTv2 module is introduced to build the backbone network, which effectively reduces parameters while ensuring the network's ability to extract features. The Efficient Layer Aggregation Network (ELAN) of the neck network is augmented with Convolutional Block Attention Module (CBAM) to broaden the receptive field of the model and enhance its attention to both the fire images channel and spatial information. Additionally, Varifocal Loss is used to address the problem of inaccurate object positioning in the edge areas of fire images. Compared with the YOLOv7-Tiny segmentation algorithm, for Box and Mask, the precision increases by 5.9% and 6.2%, recall increases by 2.5% and 3.3%, and mAP increases by 4% and 6%. In addition, the FPS reaches 55.64, satisfying the requirements for real-time detection. The improved algorithm exhibits good generalization performance and robustness.

INDEX TERMS Instance segmentation, MobileViTv2, fire detection, CBAM, varifocal loss.

I. INTRODUCTION

Fire is a serious disaster that can cause significant harm to people's safety, property, and the natural environment. If not addressed promptly, a fire can spread quickly, making it crucial to take immediate response measures in its initial stages. Currently, research on fire detection can be broadly categorized into two groups: methods based on traditional computer vision and methods based on deep learning. The traditional method for fire detection primarily analyzes color, shape, and texture features. Among these, color features are the most commonly used. However, this method is susceptible to false detections in scenes with complex backgrounds and significant variations in the lighting conditions. To overcome these limitations, some researchers have adopted a multi-feature fusion method that combines multiple features, such as color, texture, and shape, to improve detection accuracy. Li et al. [1] proposed a video-based autonomous flame detection model that utilized a Gaussian mixture color model of the Dirichlet process. Wang et al. [2] extracted features related to forest fires and developed a model for identifying fires by analyzing various features such as color, texture, and shape. Ding et al. [3] developed an Identification Flame Color Space (IFCS) model based on chaos theory and the k-medoids particle swarm optimization technique to solve high false alarm rates. To reduce false alarms and computational complexity, Khondakar et al. [4] examined color information, shape transformations, and optical flow estimation of fire while considering both static and dynamic factors, and suggested a multi-level fire detection approach. These approaches, which depend on manually extracted features for fire detection such as color, shape, and texture, are confined by preset features, resulting in significant restrictions.

In recent years, the application of deep learning in fire detection has received widespread attention. Among the various neural network architectures, the Convolutional Neural

The associate editor coordinating the review of this manuscript and approving it for publication was Yudong Zhang.

Network (CNN) is widely used to capture both local and global features in images. On the other hand, some Recurrent Neural Network (RNN) variants, such as Long Short-Term Memory (LSTM) [5] and Gated Recurrent Unit (GRU) [6], are commonly used to model and analyze fire video sequence information and extract the temporal and spatial evolution characteristics. Zhang et al. [7] proposed a fire detection model called ATT Squeeze U-Net, which utilized a U-Net network with a Squeeze Net structure. Qi et al. [8] developed a flame recognition framework designed to extract both static and dynamic features of video flame images. Li et al. [9] developed a real-time system to detect and localize indoor fires. This system can be deployed on an embedded platform, such as the Jetson Nano. Compared with traditional image processing methods, the aforementioned methods exhibit better generalization capabilities. However, fire detection using deep learning technology still encounters the following challenges.

- Collecting and labeling large-scale fire datasets is a challenging and time-consuming task, particularly when obtaining data from actual fire scenarios.
- Fires can be small, dense, large, or scattered. The ability to identify and classify a variety of fire occurrences, including those with varying shapes, sizes, and combustion characteristics, is important for fire detection algorithms.
- In fire detection, distinguishing between fire features such as flames and smoke, and the background is challenging, particularly in complex environments. This affects fire detection accuracy, leading to missed or false detections.
- Enhancing the detection capability of the model and increasing its detection efficiency are crucial for the quick and accurate detection of fire information.

In response to the above problems, this study improves the detection head, backbone network, neck network and loss function of the original YOLOv7-Tiny detection algorithm. The experimental results show that, compared to the original algorithm, the improved YOLO-SF algorithm can achieve an optimal balance between accuracy and speed. The main contributions of this study are summarized as follows.

- Label 5000 experimental images, including 3203 fire images and 1797 non-fire images, such as fire clouds and city night scenery, to improve the quality of the experimental datasets.
- The segmentation detection head of YOLOR [10] is used to jointly train and reason object detection and instance segmentation, which can provide pixel-level object position information and instance segmentation conclusions.
- To decrease the model's parameters and computational complexity, we utilize the MobileViTv2 [11] network to modify the structure of the backbone network and depth-separable convolution to replace the standard convolution.
- The Convolutional Block Attention Module (CBAM) [12] is added to the Efficient Layer Aggregation Network (ELAN) of the neck network to fully integrate feature information from the upper and lower layers of the network and improve the receptive field and ability of the model to learn global information.
- The original loss function is replaced by Varifocal Loss [13]. By adjusting the dynamic balance factors and parameters, the background weight of the fire images is reduced, and the weight of the edge, blurred area, and detailed information are increased.

The remainder of this paper is organized as follows. Section II presents related work on fire object detection and segmentation detection. The details of the improved algorithm are presented in Section III. Comparative experimental analyses of the improved algorithm and other algorithms are presented in Section IV. Finally, conclusions are presented in Section V. All code and experimental data can be accessed through the following URL: https://github.com/suyixuan123s/YOLO-SF.git.

II. RELATED WORK

Owing to the development of hardware and software technologies as well as the expansion of computer power, many researchers are using deep learning techniques for fire detection. Deep learning technology can independently extract object features from images to obtain generalization information. These approaches have superior learning abilities and adaptabilities. Common deep learning algorithms include YOLO [16], [17], [18], Faster R-CNN (region-based convolutional neural networks) [15], and SSD (single-shot multi-box detector) [14]. These algorithms combine convolutional neural networks with bounding box regression techniques to enable real-time object detection and localization, which are widely used in fire detection. Zhang et al. [15] developed a situational awareness system for investigating mine fires. Simultaneously, an optimized Faster R-CNN model is designed and implemented. Wu et al. [16], considering the unique characteristics of ship fires and marine environments, proposed a lightweight object recognition algorithm based on YOLOv4-minor. Smadi et al. [17] proposed a new framework to reduce the sensitivity of various YOLO detection models and improve the accuracy of forest smoke detection. Xu et al. [18] proposed a new flame-detection framework called YOLO-F. They replaced the neck part of YOLOv4 with FPNs-SE and used a new loss function called ACIoU to enhance the network's ability to extract features of different scales. Xue et al. [19] added the CBAM attention module to the YOLOv5 network and proposed an improved small-object forest fire detection model, which effectively solved the information loss problem caused by the small number of forest fire small object pixels. Luo et al. [20] improved YOLOX by combining the Swin Transformer architecture, CBAM attention mechanism and Slim Neck structure and proposed a flame and smoke detection algorithm applied to laboratory fires.

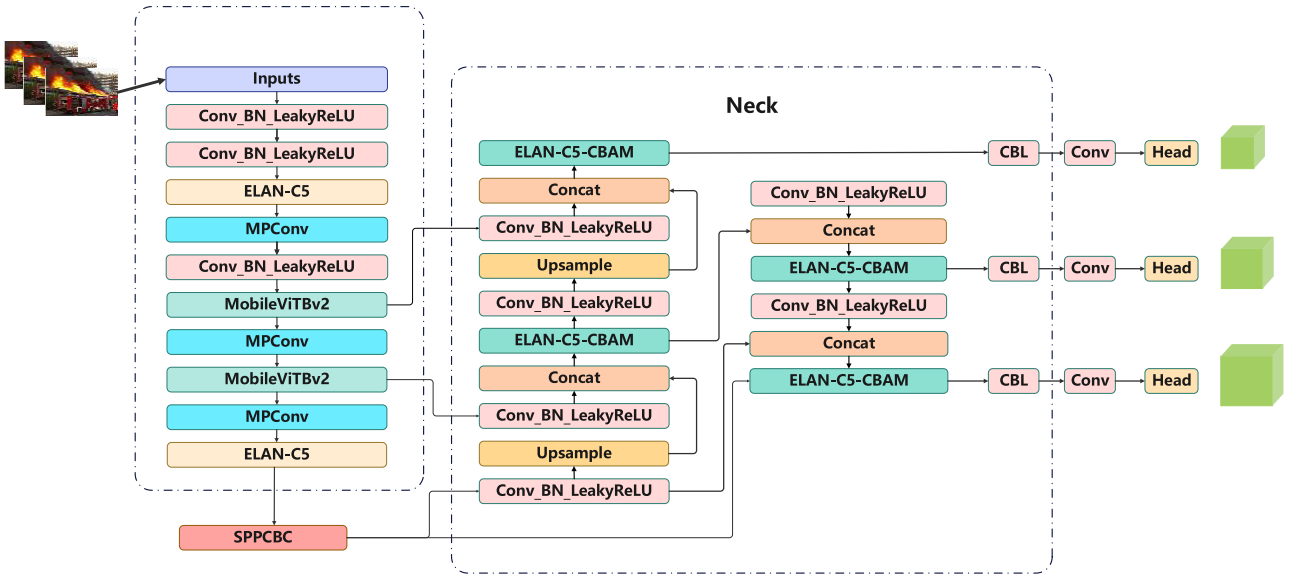


FIGURE 1. YOLO-SF network structure.

In segmentation for fire detection, each pixel of an image must be assigned to either the fire or background category. The semantic segmentation network learns the segmentation mask directly from the original image through end-to-end training. U-Net [21] and DeepLab [22] networks directly learn the semantic information of an image to achieve fire segmentation at the pixel level. For example, Zheng et al. [21] proposed a multi-level semantic segmentation method for fire smoke based on global information and U-Net network. The algorithm used Multi-Scale Residual Group Attention (MRGA) in combination with U-Net to extract multi-scale smoke features, which improved the perception ability for small-scale smoke. Harkat et al. [23] utilized the Atrous Spatial Pyramid Pooling architecture of Deeplabv3+ to improve the fire image segmentation results. They proposed a fire detection model for RGB and infrared images by integrating Xception into Deeplabv3+. Wang et al. [24] used high-sensitivity bands and remote sensing indices in the RGB, SWIR2, and AOD bands to segment smoke, and proposed a smoke segmentation network model called Smoke-U-net, which combines an extended U-net, an attention mechanism, and residual blocks. Cheng et al. [25] utilized an encoder-decoder structure with dilated separable convolutions to segment smoke regions and proposed a computer vision-based approach for smoke heat map detection. Jing et al. [26] integrated the CBAM module into the feature extraction structure to enhance the network's focus on relevant information in wildfire areas while suppressing the extraction of irrelevant features. They proposed a semantic segmentation network called Mobile-Attention-Net for efficient detection of forest fire areas.

The objective of instance segmentation is to segment each fire instance in an image into an independent object. Commonly used instance segmentation algorithms include Mask R-CNN [29], [30] and YOLACT [27]. These methods combine object detection and segmentation techniques to achieve the precise localization and segmentation of individual fire sources. For example, Sun et al. [28] proposed a semi-supervised learning method that can effectively improve the performance of fire instance segmentation by reducing segmentation errors. Guan et al. [29] proposed a method for the early detection and segmentation of forest fires based on an MS R-CNN model that used a U-shaped network to reconstruct the Mask-IoU branch. This method is referred to as Mask SU R-CNN. Zhou et al. [30] proposed a method for automatic indoor fire load detection using computer vision and a Mask R-CNN. Niu et al. [31] introduced dilated convolutions and the CBAM attention mechanism to enhance the segmentation accuracy of flame regions in images. To this end, they proposed a lightweight instance segmentation feature extraction network based on a Mask R-CNN. Martins et al. [32] used instance segmentation algorithms and the CBAM attention mechanism to extract the shape, color, and spectral features of objects. They presented a novel model for forest-fire detection. The above research results highlight the unique advantages of segmentation technology in the field of fire detection and provide an innovative direction for the further development of fire segmentation research.

III. IMPROVED YOLO-SF ALGORITHM

The YOLO-SF network structure is divided into four parts: the Input, Backbone, Neck, and YOLO Head. The network structure of the YOLO-SF is illustrated in Figure 1.

A. INPUT

The input of YOLO-SF is an RGB image, usually in standard JPEG or PNG format. Before processing, the image must be pre-processed to optimize its operation and output effects.

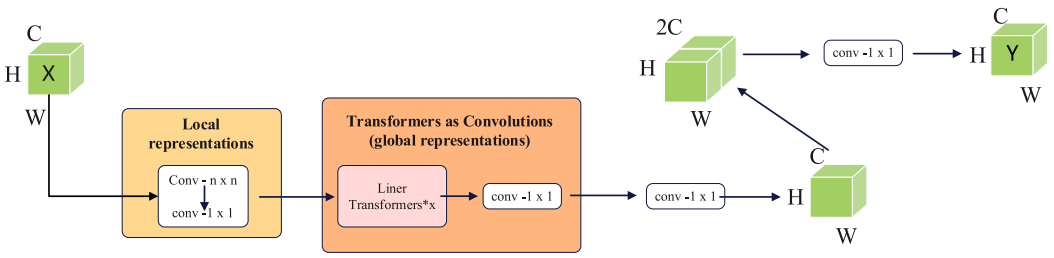


FIGURE 2. MobileViTv2 structure.

The pre-processed images are then fed into the YOLO-SF model for fire detection.

B. BACKBONE

The backbone network of YOLO-SF consists of convolutional layers, Spatial Pyramid Pooling (SPP) [33] and fully connected layers. To effectively detect fire objects of various sizes and shapes and extend the receptive field of the model, MobileViTv2, a lightweight visual transformation network, is used to transform the backbone network. It consists of two parts: a backbone network based on MobileNetV3 [34] and a head network based on a Vision Transformer (ViT) [35], [36]. The backbone network adopts lightweight, depth-separable convolutional modules and inverted residual modules to extract features from the images. This design aims to achieve efficient computation and low memory requirements. The main network is based on the concept of Vision Transformer (ViT) and uses a transformer encoder to encode and decode features to capture global features and establish the relationship between them. To further reduce the computational burden, MobileViTv2 also introduces a mechanism that considers multiple scales such that the model can prioritize features with different resolutions to improve the efficiency of the model without losing accuracy.

The structure of MobileViTv2 is illustrated in Figure 2. The dimension of the input feature layer is WHC. First, the input is routed through local representation components, using $n \times n$ convolution to encode local information, and then mapped to high dimensions by 1×1 pointwise convolution. Next, the feature information is transmitted to the global representation component. In this step, the transformer model is employed to encode global information. Subsequently, a 1×1 pointwise convolution is applied to further process mapping. In the third step, features are introduced into the fusion module, where they are mapped through 1×1 point-by-point convolution. In this step, the number of feature channels C is increased to $2C$, followed by a 1×1 pointwise convolution to normalize the information and complete the feature processing. Figure 4 provides a detailed depiction of the integration process between the backbone network and MobileViTv2.

C. NECK

The neck network of YOLO-SF incorporates the residual structure [37] and squeeze-and-excitation (SE) attention mechanism [38] to augment the network's feature extraction capability. In the neck network, the ELAN-C5 module enhances the efficiency of the feature fusion by employing stacked convolution blocks, thereby ensuring the shortest gradient path. However, this module lacks the complete integration of feature information between the upper and lower layers of the network, leading to network disregard for small objects, which hampers its capacity to capture global information. To address this issue, the CBAM attention mechanism is incorporated to handle both the channel and spatial information of features. This mechanism is integrated into the ELAN-C5 module, resulting in an ELAN-C5-CBAM architecture. Figure 3 provides a detailed depiction of the improved process.

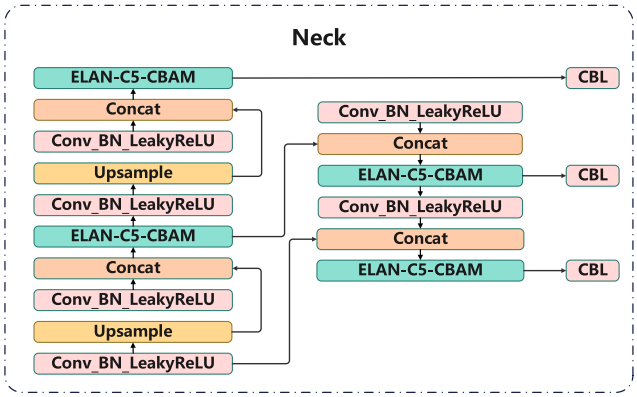


FIGURE 3. The improvement details of the Neck network.

In Figure 5, the CBAM attention module is incorporated following the C5 module of the neck network. The CBAM attention mechanism comprises two essential components: the Channel Attention Module (CAM) [12] and the Spatial Attention Module (SAM) [12]. For the feature map $F \in R^{C*H*W}$ generated by the convolutional neural network, the CBAM mechanism sequentially deduces a one-dimensional channel attention map $M_c \in R^{C*1*1}$ and two-dimensional spatial attention map $M_s \in R^{1*H*W}$. The calculation processes are shown in equation (1) and equation (2).

$$F' = M_c(F) \otimes F \tag{1}$$

$$F'' = M_s(F') \otimes F' \tag{2}$$

Among the various symbols presented, the operator $\otimes$ signifies element-wise multiplication. First, multi-channel

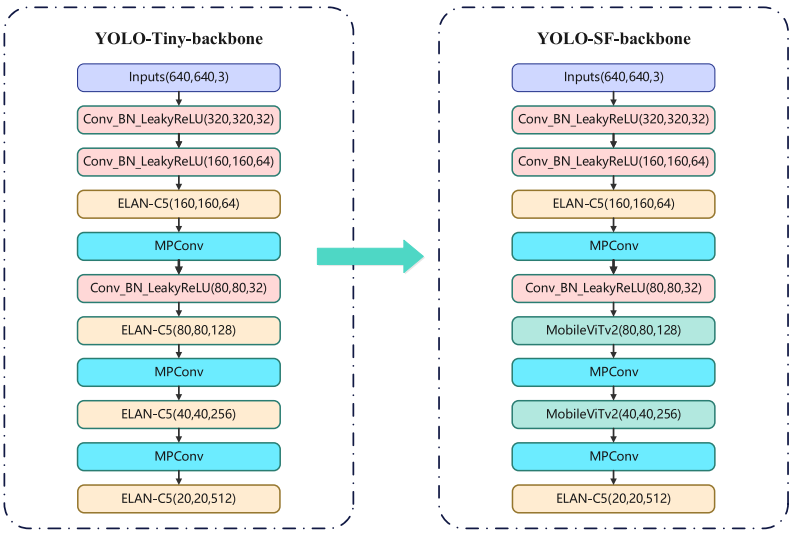


FIGURE 4. Backbone network improvement details.

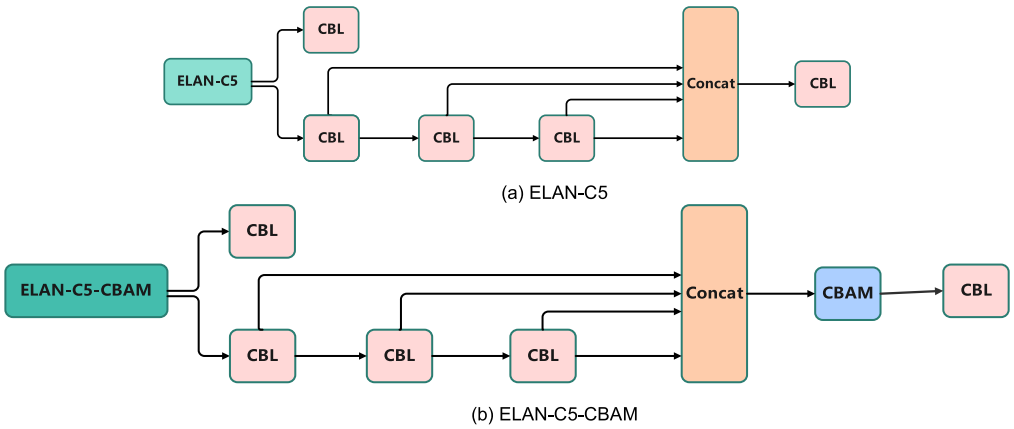


FIGURE 5. ELAN-C5-CBAM of the neck network improvement details.

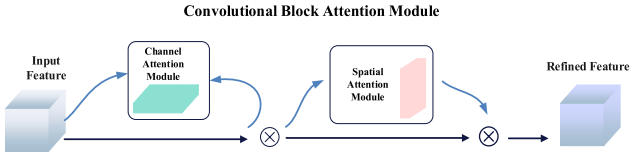


FIGURE 6. Structure of the CBAM.

attention is used to process the input feature map to derive F'. Subsequently, the processed feature map F' is dealt with spatial attention, and F'' is the output. Figure 6 depicts the operational procedure of the CBAM attention mechanism.

Each channel within the feature map is considered a feature detector. To effectively calculate the channel features, a combination of maximum pooling and average pooling is employed. The spatial dimension of the feature map is compressed, resulting in two distinct spatial background descriptions: F^c_{max} and F^c_{avg}. Then, the calculation of the channel attention map $M_c \epsilon R^{C*1*1}$ is achieved by utilizing a shared network consisting of a multi-layer perceptron (MLP). The calculation processes are described by equation (3) and equation (4).

$$M_c(F) = \alpha(MLP(AvgPool(F)) + MLP(MaxPool(F))) \tag{3}$$

$$M_c(F) = \alpha(W_1(W_0(F^c_{avg})) + W_1(W_0(F^c_{max}))) \tag{4}$$

Among these variables, $W_0 \in R^{\frac{C}{r}*C}$ and $W_1 \in R^{C*C/r}$ are distinct. Unlike channel attention, spatial attention primarily emphasizes location-related information. To compute the spatial features, the channel dimensions underwent maximum and average pooling operations. These operations yield two distinct feature descriptions, denoted as $F^s_{max} \in R_{1*H*W}$ and $F^s_{avg} \in R_{1*H*W}$. Subsequently, a densely connected layer is employed to merge the two distinct feature descriptions and execute the convolution operation, thereby generating a spatial attention map $M_s(F) \in R_{H*W}$. The calculation

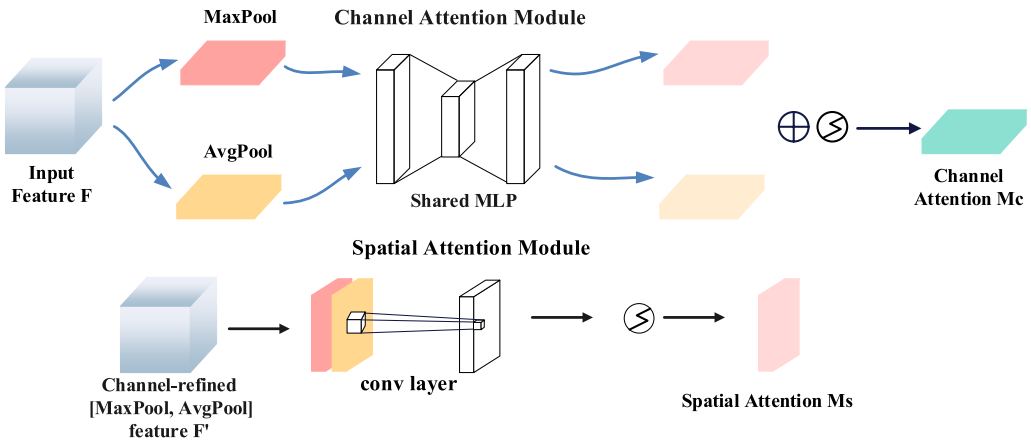


FIGURE 7. Structure of CAM and SAM.

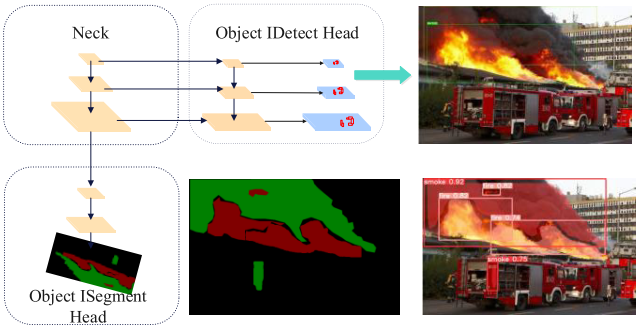


FIGURE 8. Detection head of YOLO-SF.

processes are shown in equation (5) and equation (6). Figure 7 shows the working principles of channel attention and spatial attention.

$$M_s(F) = \alpha(f^{7*7}([AvgPool(F); MaxPool(F)])) \quad (5)$$

$$M_s(F) = \alpha(f^{7*7}([F^s_{avg}; F^s_{max}])) \quad (6)$$

D. YOLO HEAD

During fire incidents, it is common for flames, smoke, and various objects to obstruct and intertwine with one another. The original YOLOv7-Tiny object detection algorithm can only provide a bounding box that surrounds the entire object area, but cannot provide pixel-level object boundary information. In particular, in the case of occlusion or overlap, localization of the fire area becomes more inaccurate. In addition, owing to the difficulty in obtaining detailed information about the object area, the processing of small-object fires has significant limitations.

To address these issues, this study carries out objective improvements to the detection head of YOLOv7-Tiny. Specifically, the object detection head in YOLOv7-Tiny is replaced with the segmentation detection head of YOLOR, which realizes the integration of object detection and pixel-level segmentation. YOLOR's segmentation detection head is a crucial component that combines object detection and segmentation tasks. It defines an 'IDetect' class for object detection tasks and an 'ISegment' class for segmentation tasks. The 'ISegment' class inherits from and extends the 'IDetect' class to process the segmentation task. This inheritance and extension approach allows the creation of models for different vision tasks without re-writing shared functionality [39].

First, the 'IDetect' class utilizes the fire image features extracted by the backbone and neck networks to generate candidate bounding boxes for the identified objects and predicts the object category and confidence level. Next, the 'ISegment' class generates a segmentation mask for each candidate box based on shared features, classifying image pixels as foreground objects or background regions. Segmentation masks make pixel-level predictions of fire and smoke and segment fire objects from images. Finally, the object detection results are integrated with pixel-level segmentation masks to yield location information and pixel-level segmentation outcomes for each fire object. The distinctive design of the improved head allows simultaneous achievement of object location and pixel-level segmentation in fire image detection. This capability enables a more comprehensive and accurate analysis of fire information. Figure 8 shows the detection process after adding the segmentation detection head.

E. VARIFOCAL LOSS

Fire objects usually occupy only a small portion of the entire detection image, whereas the background occupies most of the space. This phenomenon leads to an imbalance in the fire categories. Additionally, accurately classifying the edges of flames or blurred pixels is challenging. In this case, the original loss function may allocate excessive weight to the background pixels or equal weights to all pixels, making it difficult for the model to accurately detect and segment

fire objects. This study utilizes Varifocal Loss to address these issues.

Varifocal Loss introduces dynamic focusing factors that can be dynamically adjusted based on the complexity, prediction probability, and confidence of the fire samples. This allows for a more precise and balanced focus on the fire categories. The introduction of a boundary-aware loss function can prioritize the boundary region of an object. By incorporating a supplementary loss term into the bounding region, the regression of the bounding box can be effectively optimized, leading to an enhancement in its localization accuracy. The concept of Varifocal Loss is derived from Focal Loss [45], [46]. The mathematical expression for the Focal Loss is shown in equation (7).

$$FL\left(p_t\right) = -\left(1-p_t\right)^{\gamma}\log\left(p_t\right), \quad p_t = \begin{cases} p, y=1 \\ 1-p, y=0 \end{cases} \tag{7}$$

$$VFL\left(p_t\right) = -\alpha(1-p_t)^{\gamma}\log\left(p_{t\_hat}\right) - \alpha(1-p_{t-hat})^{\beta}\log(p_{t\_hat}) \tag{8}$$

Equation (8) is the mathematical expression of Varifocal Loss, among these variables, the p_t is the probability that the model predicts that the sample is positive class, the p_{t_hat} is the probability that the model predicts that the sample is positive class, and the prediction is correct, the α is a balance factor, used to adjust the weights between positive and negative samples. When the predicted probability approaches one, the focal factor exhibits a gradual decrease, whereas when the predicted probability approaches zero, the focal factor increases. The γ and β are adjustable parameters used to adjust the weights of easily and accurately classified samples. The incorporation of these parameters can improve the focus accuracy and promote balance, thereby optimizing the capacity of the model to detect and segment fire objects.

IV. EXPERIMENTS AND RESULT ANALYSIS

A. EXPERIMENTAL PREPARATION

1) DATASET

Currently, publicly accessible fire datasets are scarce. During the data collection process, duplicate fire images are excluded to guarantee the diversity and uniqueness of the collected data. We utilize LabelMe software to perform data labeling and name it FSD. During data-labeling process, the images are categorized into two groups: smoke and fire.

Subsequently, they are further divided into a training set and test set at a ratio of 9:1. Additionally, a portion of the training set, specifically 10%, is reserved as the validation set. The dataset encompasses a variety of scenarios, including grasslands, forests, buildings, roads, and small-scale fires. Figure 9 shows a section of the fire segmentation images. The URL dataset can be obtained from https://github.com/suyixuan123s/Fire-Segmentation-Dataset.git.

A total of 3203 fire images and 1797 non-fire images are labeled. These non-fire images encompass various visual images, such as sunsets and city lights, thereby enhancing the diversity of the dataset. Figure 10 shows some negative sample images. It is evident from Figure 11 that the segmentation dataset contains approximately 6000 flame labels and 3000 smoke labels. The distribution and proportion of labeled data exhibit both evenness and diversity.

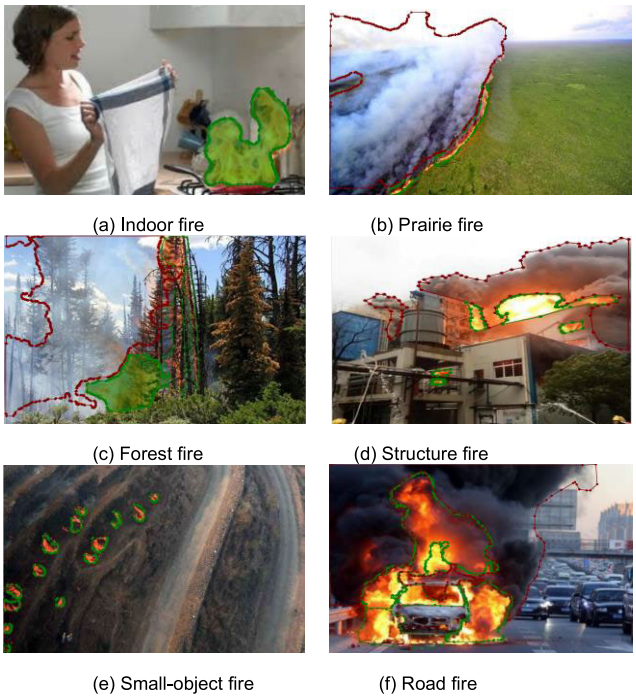


FIGURE 9. Annotation flow of datasets and scenarios.

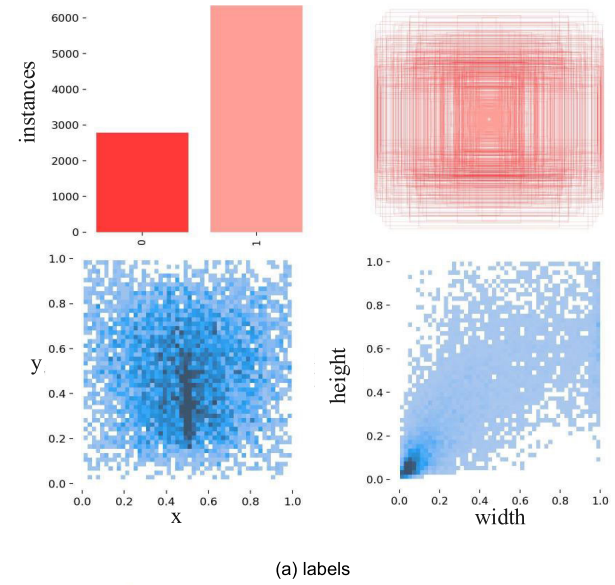
FIGURE 10. Some negative sample examples.

2) EXPERIMENTAL ENVIRONMENT AND PARAMETER SETTING

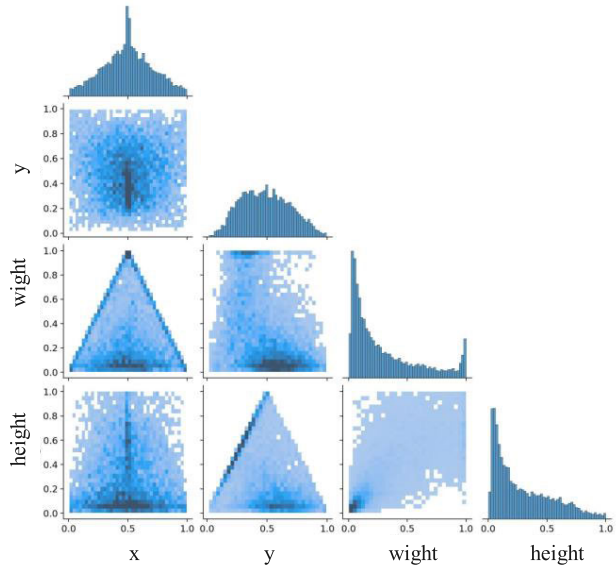
The configuration of the experimental environment is listed in Table 1. The input image size is 640×640, with 300 epochs and a batch size of 64. The optimizer is SGD with a patience of 100. The mosaic value is set to 1.0, lr0 to 0.01, momentum to 0.937, and weigh-decay to 0.0005.

3) EVALUATION INDICATORS

In the experimental analysis, we evaluate two output methods: Box and Mask. We introduce the following metrics to



(a) labels



(b) Label correlations

FIGURE 11. Experimental data visualization.

assess the algorithms used in this study: Precision (Box), Precision (Mask), Recall (Box), Recall (Mask), F1 (Box), F1 (Mask), mAP (Box), mAP (Mask), FPS, and Parameters. TP (Box) indicates the number of instances that are correctly detected as objects, FP (Box) indicates the number of instances that are incorrectly detected as objects, and FN (Box) indicates the number of actual object instances that are not detected. TP (Mask) denotes the number of pixels that have been accurately classified as objects. FP (Mask) refers to the number of pixels erroneously classified as objects. FN (Mask) represents the number of the actual object pixels that have not been correctly classified.

The number of object categories is denoted as n_{class}, where n_{recall} represents the number of set recall thresholds.

TABLE 1. Environment configuration.

Schedule	Capacity
Parameters Configuration	Operating system AutoDL Server Linux Operating system
CPU	12 vCPU Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50GHz
GPU	V100-SXM2-32GB (32GB) * 1
RAM	43GB
Deployment Environment	Python3.8.10
Deep learning framework	PyTorch1.11.0
Accelerated Computing Architecture	CUDA11.3

The average precision of a single category at a given recall rate r is denoted as $AP_{Box}(r)$和$AP_{Mask}(r)$. These evaluation indicators are commonly referred to as P(B), P(M), R(B), R(M), F1(B), F1(M), mAP(B), mAP(M), FPS, and Params. The mathematical expressions are shown in equations (9)- (16).

$$Precision(Box) = \frac{TP(Box)}{(TP(Box) + FP(Box)} \tag{9}$$

$$Precision(Mask) = \frac{TP(Mask)}{(TP(Mask) + FP(Mask)}10) \tag{10}$$

$$Recall(Box) = \frac{TP(Box)}{(TP(Box) + FN(Box)} \tag{11}$$

$$Recall(Box) = \frac{TP(Mask)}{(TP(Mask) + FN(mask)} \tag{12}$$

$$F1(Box) = \frac{2 \times Precision(Box) + Recall(Box)}{Precision(Box) \times Recall(Box)} \tag{13}$$

$$F1(Box) = \frac{2 \times Precision(Mask) + Recall(Mask)}{Precision(Mask) \times Recall(Mask)} \tag{14}$$

$$mAP\,(Box) = \frac{1}{n_{class}} \sum_{i=1}^{n_{class}} \frac{1}{n_{recall}} \sum_{r=0}^{n_{recall}} AP_{Box}(r) \tag{15}$$

$$mAP\,(Mask) = \frac{1}{n_{class}} \sum_{i=1}^{n_{class}} \frac{1}{n_{recall}} \sum_{r=0}^{n_{recall}} AP_{Mask}(r) \tag{16}$$

The parameters are employed to quantify the scale of a model, denoting the adjustable variables that must be acquired during the learning process of the model. These parameters directly affect the storage capacity and computational complexity of a model. FPS is used to measure the processing speed of the model for real-time images and videos. This represents the number of frames processed per second. When the FPS value is greater than 20, the requirements for real-time detection are satisfied.

B. COMPARISON EXPERIMENTS

1) COMPARISON EXPERIMENT FOR THE IMPROVEMENT OF THE BACKBONE NETWORK

Enhancing model performance is a commonly employed strategy by utilizing various network modules to optimize

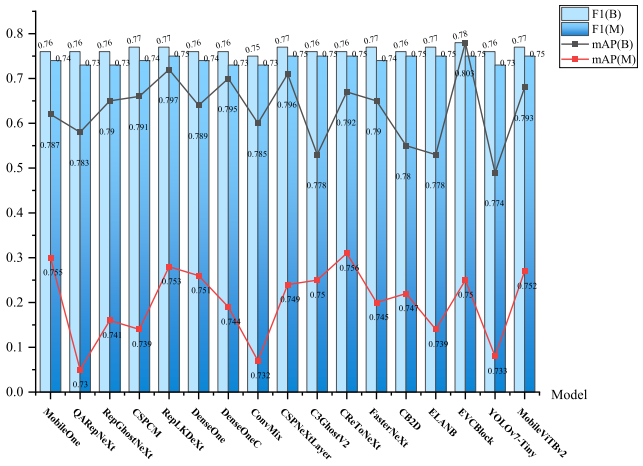


FIGURE 12. Comparison of F1(B), F1(M), mAP(B), and mAP(M) results of the enhanced backbone network.

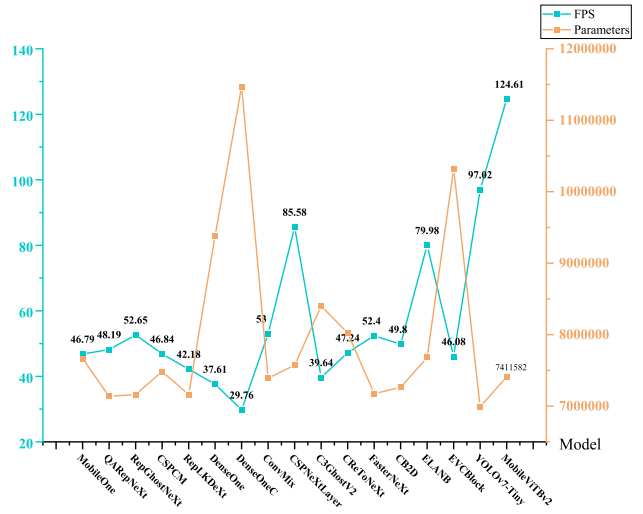


FIGURE 13. Comparison of FPS and Parameters results of the improved backbone network.

the backbone network. From Figure 12 and Figure 13, it is evident that the modules with significant improvement effects are the RepLKDeXt [40], CSPNeXtLayer, EVCBlock [41], CReToNeXt [42], ELANB, and MobileViTv2. The MobileViTv2 module exhibits significant improvement in the detection effect, with its mAP(B) and mAP(M) achieving 0.803 and 0.752 respectively, representing the global optimal values. In addition, the F1(B) and F1(M) are 0.780 and 0.750, respectively. The FPS reaching 124.61 satisfies the real-time detection requirements, the parameters exhibit a decrease. This demonstrates that the network structure of MobileViTv2 can significantly enhance the network's fire detection capability.

2) COMPARATIVE EXPERIMENT TO IMPROVE THE ATTENTION MECHANISM

Based on the improvement in the backbone network, different attention mechanisms are incorporated into the neck network, and a series of comparative experiments are conducted. As shown in Figure 14, after combining ECA [43], SimAM [44], and CBAM, the detection effect is significantly improved. After adding the CBAM module, the mAP(B) and mAP(M) reach 0.811 and 0.756 respectively, which are the global optimal levels. This can be attributed to the adaptive nature of the CBAM attention mechanism, which can effectively adjust the weight distribution of the feature maps and focus more on fire-related features, thereby significantly improving the representation ability of the model.

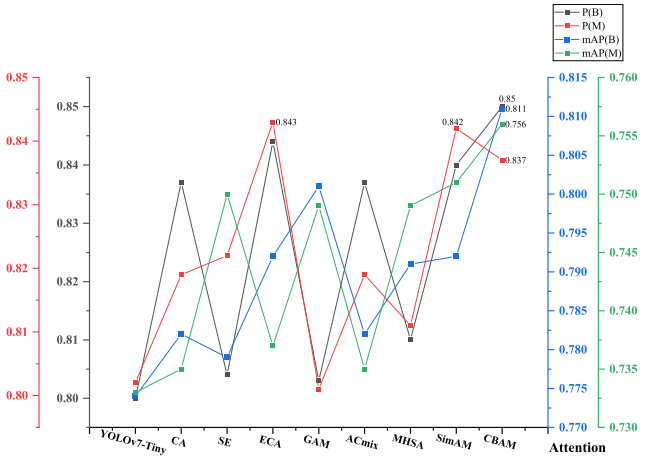


FIGURE 14. Comparison of results of the attention mechanism improvement.

To determine the optimal placement of the CBAM module, this study refers to the improvement strategy of the CBAM module and proposes three improvement schemes to verify the most effective position for the CBAM module.

1. The CBAM modules are incorporated after the connection layer of the C5 module within the backbone network.
2. The CBAM modules are incorporated after the connection layer of the neck network's C5 module.
3. The CBAM modules are incorporated after the connection layer of the C5 module in the backbone and neck networks.

From the experimental data presented in Table 2, it is evident that the incorporation of the CBAM module significantly enhances the detection performance of the model. The detection outcome in scheme two demonstrates superior performance. It has been verified that the addition of a CBAM module to the neck network can significantly enhance its effectiveness.

3) COMPARATIVE EXPERIMENT TO LOSS FUNCTION

The training direction of the model can be guided using an appropriate loss function. After completion of the backbone network, neck network, and detection head modification, an experimental evaluation is carried out to assess the performance of four loss functions: Quality Focal loss [45], Poly loss [46], Focal Loss, and Varifocal Loss.

Figure 15 clearly shows that using Varifocal Loss results in a significant increase in R(B) and R(M), obtaining optimal

TABLE 2. The CBAM module improvement experiment.

Num	Class	P(B)	P(M)	R(B)	R(M)	mAP(B)	mAP(M)	FPS	Params (MB)
N1	all	0.833	0.832	0.727	0.681	0.785	0.740	62.15	43.21
N2	all	0.850	0.837	0.732	0.700	0.811	0.756	66.86	47.71
N3	all	0.826	0.826	0.722	0.667	0.790	0.740	55.64	62.70

TABLE 3. Comparison experiment with other algorithms.

Model	Class	P(B)	P(M)	R(B)	R(M)	Class	mAP(B)	mAP(M)	FPS	Params
YOLOv5(X)$^{(S)}$	smoke	0.901	0.874	0.732	0.705	all	0.785	0.771	60.46	336.82
	fire	0.821	0.821	0.662	0.658					
YOLOv7$^{(S)}$	smoke	0.879	0.865	0.778	0.766	all	0.807	0.796	77.07	144.38
	fire	0.811	0.805	0.662	0.658					
YOLOv7(x)$^{(S)}$	smoke	0.906	0.910	0.762	0.773	all	0.800	0.793	73.73	299.36
	fire	0.795	0.784	0.668	0.667					
YOLOv8(n)$^{(S)}$	smoke	0.858	0.863	0.744	0.747	all	0.777	0.775	113.63	12.43
	fire	0.800	0.793	0.665	0.659					
YOLOv8(x)$^{(S)}$	smoke	0.904	0.895	0.787	0.776	all	0.797	0.794	50.67	273.60
	fire	0.743	0.747	0.650	0.649					
YOLACT [27]	smoke		0.883		0.749	all		0.786	31.26	129.87
	fire		0.761		0.648					
Mask R CNN [29]	smoke		0.870		0.755	all		0.790	33.69	62.00
	fire		0.789		0.665					
PSP-Net	smoke		0.839		0.727	all		0.775	52.32	58.65
	fire		0.746		0.634					
H-Net	smoke		0.864		0.758	all		0.762	15.71	323.83
	fire		0.761		0.640					
Deeplabv3-plus [22]	smoke		0.896		0.762	all		0.796	47.16	136.62
	fire		0.807		0.654					
U-Net [21]	smoke		0.870		0.755	all		0.773	51.29	34.49
	fire		0.786		0.637					
YOLOv7-Tiny$^{(S)}$	smoke	0.798	0.810	0.750	0.721	all	0.774	0.733	97.02	26.67
	fire	0.801	0.794	0.682	0.628					
YOLOv7-Tiny$^{(D)}$	smoke	0.910		0.712		all	0.772		90.91	17.05
	fire	0.627		0.737						
YOLO-SF	smoke	0.916	0.913	0.760	0.760	all	0.814	0.793	55.64	58.41
	fire	0.802	0.815	0.721	0.654					

global values of 0.741 and 0.707, respectively. In addition, the mAP(B) and mAP(M) achieve 0.814 and 0.793, respectively. The results indicate that the performance of the fire detection network can be improved through an appropriate loss function.

4) COMPARISON OF LOSS CURVE RESULTS DURING TRAINING

As shown in Figure 16, as the number of training epochs increases, the loss curves exhibit a consistent downward trend and eventually reach a stable state. When the epoch reaches 300, the model gradually approaches a state of convergence, and there is no indication of overfitting throughout the training process. Compared with the YOLOv7-Tiny algorithm, the improved YOLO-SF exhibits a lower training loss and a more pronounced downward trend, indicating its superior fitting capability. This result confirms the efficacy and robustness of the proposed algorithm.

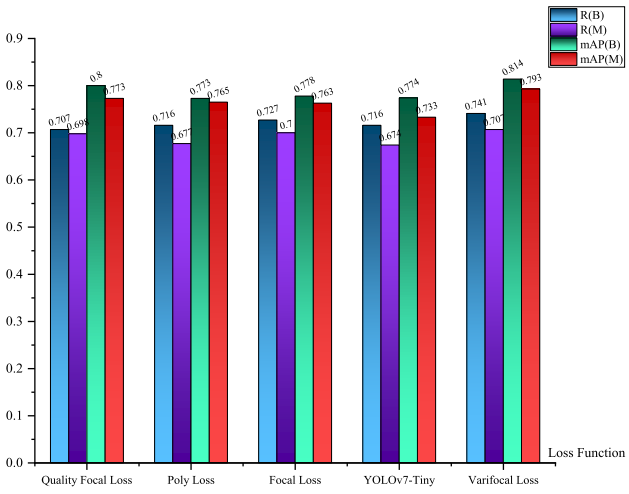


FIGURE 15. Comparison of results of loss function improvement.

5) COMPARISON EXPERIMENTS WITH OTHER ALGORITHMS

It can be observed from Table 3 that the indicators for smoke detection are higher than those for flame detection. The YOLO-SF algorithm achieves better results than the other algorithms. Compared to the object detection algorithm of YOLOv7-Tiny, each box indicator reveals significant

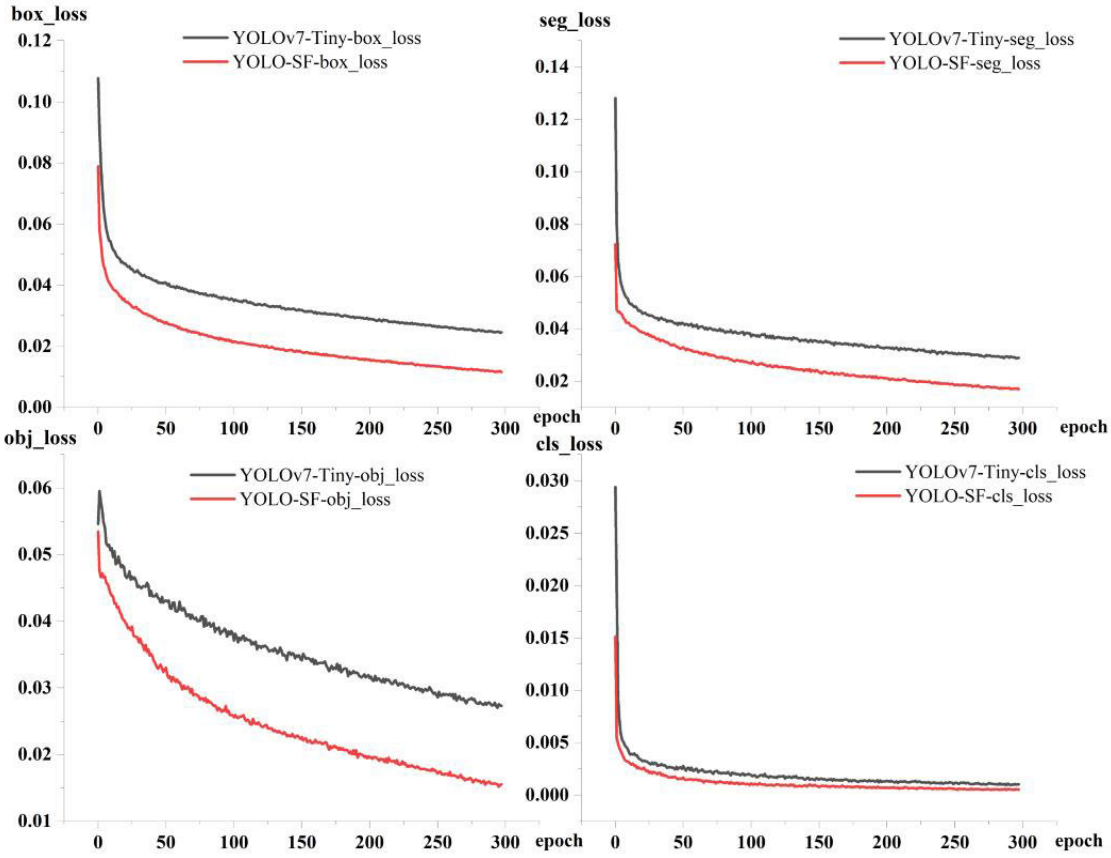


FIGURE 16. Comparison of training loss results between YOLOv7-Tiny and YOLO-SF.

improvements. Compared to the YOLOv7-Tiny segmentation algorithm, the smoke detection precision is improved by 0.118 and 0.103, and the recall is increased by 0.010 and 0.039, respectively. The flame detection precision is enhanced by 0.001 and 0.019, recall is improved by 0.039 and 0.026, and the mAP is increased by 0.040 and 0.060, respectively. Compared with various segmentation algorithms in literature such as YOLACT, Deeplabv3-plus, U-Net, Mask R CNN, H-Net and various YOLO segmentation algorithms, the YOLO-SF algorithm proposed in this paper has obtained better detection results. It is noteworthy that despite the increase in the model parameters, the FPS achieves 55.64, still surpassing the threshold of 20 and, meeting real-time detection requirements. The results demonstrate that the YOLO-SF algorithm presented in this study can effectively balance detection accuracy and efficiency.

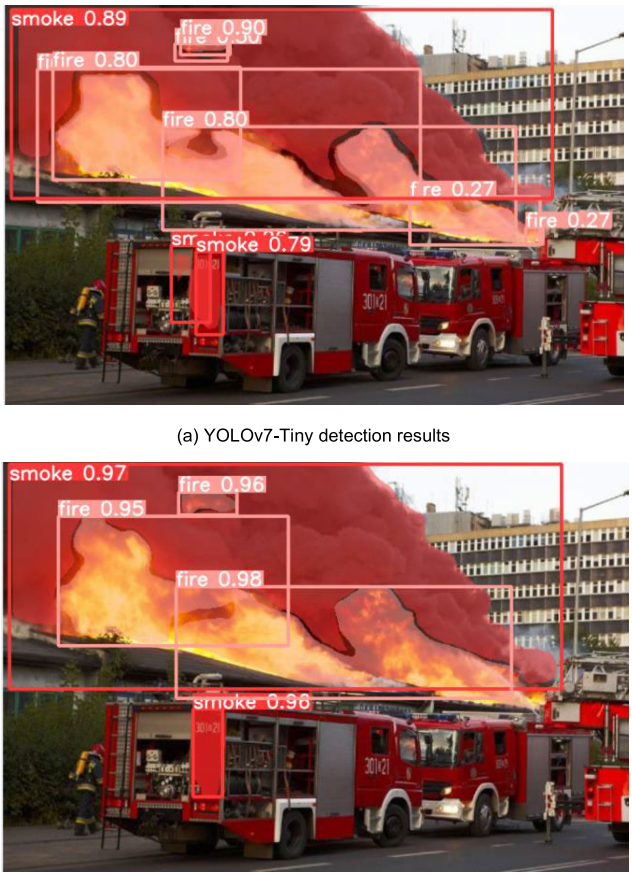
C. THE ABLATION LABORATORY

To assess the influence of each improvement factor on network performance, a series of ablation experiments are conducted. In the experiments, the environment and parameter settings are identical. The experimental results are presented in Table 4. By implementing ablation experiments, we assess the individual impact of each improvement point on the algorithm's performance and gain a comprehensive understanding of its role in enhancing the overall performance.

The first experiment is the results of the YOLOv7-Tiny object detection algorithm. Because this method lacks a segmentation function, the mask metrics are empty. The experiment serves as a reference for four subsequent experiments. In the second experiment, the segmentation detection head of YOLOR is utilized. The P(B) increase 0.032, R(B) increase 0.012, and mAP(B) increase 0.011. The third experiment employs the MobileViTv2 module to adjust the backbone network architecture. The results indicate that in comparison to experiment two, there is an increase of 0.041 in P(B) and 0.026 in P(M). Additionally, R(B) and R(M) increase 0.021 and 0.018 respectively, while mAP(B) and mAP(M) increase 0.019. The CBAM attention module is used in the fourth experiment. Compared to experiment three, the P(B) and P(M) increase 0.009 and 0.011, respectively. Additionally, the mAP(B) and mAP(M) increase 0.018 and 0.004, respectively. The fifth experiment adopts Varifocal Loss. Compared to the fourth experiment, the P(B) and P(M) increase 0.009 and 0.027, respectively, the recall R(B) and R(M) increase 0.009 and 0.007, respectively. Furthermore, the mAP(B) and mAP(M) increase 0.003 and 0.001, respectively. The addition of various improved modules has led to continuous improvement in fire detection results.

TABLE 4. Ablation experiment.

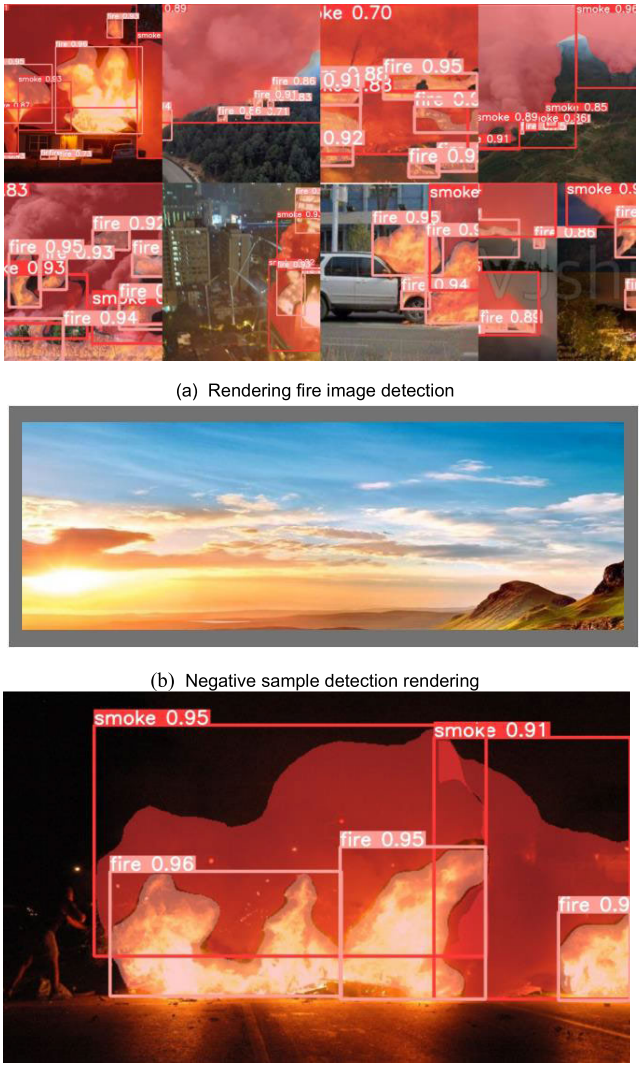
Num	Class	P(B)	P(M)	R(B)	R(M)	mAP(B)	mAP(M)	FPS	Params
N1		0.768		0.704		0.763		90.91	17.05
N2	Head	0.800	0.802	0.716	0.674	0.774	0.733	97.02	26.67
N3	Head+v2	0.841	0.828	0.737	0.692	0.793	0.752	124.61	28.27
N4	Head+v2+CBAM	0.850	0.837	0.732	0.700	0.811	0.756	66.86	47.71
N5	Head+v2+CBAM+VFL	0.859	0.864	0.741	0.707	0.814	0.793	55.64	58.41



(a) YOLOv7-Tiny detection results

(b) YOLO-SF detection results

FIGURE 17. Comparison of YOLOv7-Tiny and YOLO-SF verification results.



(a) Rendering fire image detection

(b) Negative sample detection rendering

(c) Fire video detection rendering

FIGURE 18. Fire scene detection in actual cases.

Compared with the original YOLOv7-Tiny object detection and segmentation model, the Precision, Recall, and mAP of YOLO-SF are all improved. However, the parameters increase, and the FPS decrease. The analysis of these differences is mainly due to the following factors. First, the introduction of the segmentation detection head adds additional network layers and computational complexity to achieve a pixel-level object segmentation. Second, the CBAM module introduces additional convolutional layers and attention weight parameters for the channel and spatial attention calculations, which increases the parameters. Although the FPS has decreased, it remains above 20, satisfying the requirements for real-time detection. While the parameters have increased, it is essential to emphasize the significance of achieving a higher detection accuracy in practical applications, particularly in the context of fire monitoring. Through the optimization of hardware and the rational allocation of computing resources, it is possible to improve the FPS, mitigate the impact of parameter increases, and achieve a balance between accuracy and speed, thereby ensuring the effective application of YOLO-SF in real scenarios.

D. THE PRACTICAL LABORATORY

For the detection of the same fire scene, the YOLO-SF algorithm demonstrates higher detection results than the original YOLOv7-Tiny segmentation algorithm. In Figure 17, the detection results of YOLO-SF exceed 95%, verifying that the improved network has superior detection performance.

To verify the practicality of the YOLO-SF algorithm further, several real-life cases are tested, as shown in Figure 18. The results of the YOLO-SF algorithm for detecting flames and smoke are approximately 90%, which can effectively avoid the false detection of fire negative sample images. The above experiments prove that the improved algorithm has obvious robustness and practicability in fire detection, which provides strong support for its application in actual scenarios.

V. CONCLUSION

The updated YOLO-SF algorithm can accurately recognize fire objects and their backgrounds, regardless of the magnitude or type of fire scene, and can effectively detect and locate fires. These results demonstrate the superior performance of the proposed algorithm in terms of its accuracy, robustness, stability, and practicality. Despite significant improvements, we acknowledge that the YOLO-SF algorithm still has room for improvement in terms of fire detection recall. This requires further research and fine-tuning.

We will focus on enhancing recall to further improve the performance of the algorithm. This could entail screening and strengthening of more samples for difficult-to-detect fire objects, as well as refining the network topology and loss function to better capture and reflect fire object features. Furthermore, we will investigate the application of the algorithm to more difficult environmental situations, such as nighttime fire detection, rain, snow, and other severe weather conditions. These improvements improve the applicability and utility of the proposed algorithm.

These models have the potential to become significant tools for early fire detection, thereby enhancing firefighter safety, improving the monitoring of buildings and infrastructure, enabling the implementation of automated fire suppression systems, and optimizing future emergency response operations. These improvements will lead to important innovations in fire prevention and disaster management. We hope to reduce the damage and loss caused by fire, improve the safety level of society and the ability to deal with disasters, and provide necessary technical support for future fire prevention and emergency response.

REFERENCES

[1] Z. Li, L. S. Mihaylova, O. Isupova, and L. Rossi, "Autonomous flame detection in videos with a Dirichlet process Gaussian mixture color model," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 1146–1154, Mar. 2018, doi: 10.1109/TII.2017.2768530.

[2] Y. Wang, L. Dang, and J. Ren, "Forest fire image recognition based on convolutional neural network," *J. Algorithms Comput. Technol.*, vol. 13, Nov. 2019, Art. no. 1748302619887689, doi: 10.1177/1748302619887689.

[3] X. Ding and J. Gao, "A new intelligent fire color space approach for forest fire detection," *J. Intell. Fuzzy Syst., Appl. Eng. Technol.*, vol. 42, no. 6, pp. 5265–5281, Apr. 2022.

[4] A. Khondaker, A. Khandaker, and J. Uddin, "Computer vision-based early fire detection using enhanced chromatic segmentation and optical flow analysis technique," *Int. Arab J. Inf. Technol.*, vol. 17, no. 6, pp. 947–953, Nov. 2020, doi: 10.34028/iajit/17/6/13.

[5] B. Kim and J. Lee, "A Bayesian network-based information fusion combined with DNNs for robust video fire detection," *Appl. Sci.*, vol. 11, no. 16, p. 7624, Aug. 2021, doi: 10.3390/app11167624.

[6] M. Pan, H. Zhou, J. Cao, Y. Liu, J. Hao, S. Li, and C.-H. Chen, "Water level prediction model based on GRU and CNN," *IEEE Access*, vol. 8, pp. 60090–60100, 2020, doi: 10.1109/ACCESS.2020.2982433.

[7] J. Zhang, H. Zhu, P. Wang, and X. Ling, "ATT squeeze U-Net: A lightweight network for forest fire detection and recognition," *IEEE Access*, vol. 9, pp. 10858–10870, 2021, doi: 10.1109/ACCESS.2021.3050628.

[8] R. Qi and Z. Liu, "Extraction and classification of image features for fire recognition based on convolutional neural network," *Traitement Signal*, vol. 38, no. 3, pp. 895–902, Jun. 2021, doi: 10.18280/ts.380336.

[9] Y. Li, J. Shang, M. Yan, B. Ding, and J. Zhong, "Real-time early indoor fire detection and localization on embedded platforms with fully convolutional one-stage object detection," *Sustainability*, vol. 15, no. 3, p. 1794, Jan. 2023, doi: 10.3390/su15031794.

[10] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "You only learn one representation: Unified network for multiple tasks," 2021, *arXiv:2105.04206*.

[11] S. Mehta and M. Rastegari, "Separable self-attention for mobile vision transformers," 2022, *arXiv:2206.02680*.

[12] Z. Li, B. Li, H. Ni, F. Ren, S. Lv, and X. Kang, "An effective surface defect classification method based on RepVGG with CBAM attention mechanism (RepVGG-CBAM) for aluminum profiles," *Metals*, vol. 12, no. 11, p. 1809, Oct. 2022, doi: 10.3390/met12111809.

[13] Z. Shao, H. Lyu, Y. Yin, T. Cheng, X. Gao, W. Zhang, Q. Jing, Y. Zhao, and L. Zhang, "Multi-scale object detection model for autonomous ship navigation in maritime environment," *J. Mar. Sci. Eng.*, vol. 10, no. 11, p. 1783, Nov. 2022, doi: 10.3390/jmse10111783.

[14] R. Bohush and N. Brouka, "Smoke and flame detection in video sequences based on static and dynamic features," in *Proc. Signal Process., Algorithms, Architectures, Arrangements, Appl. (SPA)*, Poznań, Poland, Sep. 2013, pp. 20–25.

[15] J. Zhang, Y. Jia, D. Zhu, W. Hu, and Z. Tang, "Study on the situational awareness system of mine fire rescue using faster Ross Girshick-convolutional neural network," *IEEE Intell. Syst.*, vol. 35, no. 1, pp. 54–61, Jan. 2020, doi: 10.1109/MIS.2019.2943850.

[16] H. Wu, Y. Hu, W. Wang, X. Mei, and J. Xian, "Ship fire detection based on an improved YOLO algorithm with a lightweight convolutional neural network model," *Sensors*, vol. 22, no. 19, p. 7420, Sep. 2022, doi: 10.3390/s22197420.

[17] Y. Al-Smadi, M. Alauthman, A. Al-Qerem, A. Aldweesh, R. Quaddoura, F. Aburub, K. Mansour, and T. Alhmiedat, "Early wildfire smoke detection using different YOLO models," *Machines*, vol. 11, no. 2, p. 246, Feb. 2023, doi: 10.3390/machines11020246.

[18] K. Xu, Y. Xu, Y. Xing, and Z. Liu, "YOLO-F: YOLO for flame detection," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 37, no. 1, Jan. 2023, Art. no. 2250043.

[19] Z. Xue, H. Lin, and F. Wang, "A small target forest fire detection model based on YOLOv5 improvement," *Forests*, vol. 13, no. 8, p. 1332, Aug. 2022, doi: 10.3390/f13081332.

[20] M. Luo, L. Xu, Y. Yang, M. Cao, and J. Yang, "Laboratory flame smoke detection based on an improved YOLOX algorithm," *Appl. Sci.*, vol. 12, no. 24, p. 12876, Dec. 2022, doi: 10.3390/app122412876.

[21] Y. Zheng, Z. Wang, B. Xu, and Y. Niu, "Multi-scale semantic segmentation for fire smoke image based on global information and U-Net," *Electronics*, vol. 11, no. 17, p. 2718, Aug. 2022, doi: 10.3390/electronics11172718.

[22] X. Zhang, H. Bian, Y. Cai, K. Zhang, and H. Li, "An improved tongue image segmentation algorithm based on Deeplabv3+ framework," *IET Image Process.*, vol. 16, no. 5, pp. 1473–1485, Apr. 2022.

[23] H. Harkat, J. M. P. Nascimento, and A. Bernardino, "Fire detection using residual Deeplabv3+ model," in *Proc. Telecoms Conf. (ConfTELE)*, Leiria, Portugal, Feb. 2021, pp. 1–6, doi: 10.1109/ConfTELE50222.2021.9435459.

[24] Z. Wang, P. Yang, H. Liang, C. Zheng, J. Yin, Y. Tian, and W. Cui, "Semantic segmentation and analysis on sensitive parameters of forest fire smoke using smoke-UNet and Landsat-8 imagery," *Remote Sens.*, vol. 14, no. 1, p. 45, Dec. 2021, doi: 10.3390/rs14010045.

[25] S. Cheng, J. Ma, and S. Zhang, "Smoke detection and trend prediction method based on Deeplabv3+ and generative adversarial network," *Proc. SPIE*, vol. 28, no. 3, pp. 33001–33006, 2019, doi: 10.1117/1.JEI.28.3.033006.
[26] K. Jing, Y. Jia, C. Zhang, and Z. Qin, "MobileAttentionNet: An efficient network for semantic segmentation of forest fire images," in *Proc. 6th Int. Symp. Comput. Inf. Process. Technol. (ISCIPT)*, Changsha, China, Jun. 2021, pp. 377–380, doi: 10.1109/ISCIPT53667.2021.00082.
[27] J. Zeng, H. Ouyang, M. Liu, L. U. Leng, and X. Fu, "Multi-scale YOLACT for instance segmentation," *J. King Saud Univ., Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9419–9427, 2022.
[28] G. Sun, Y. Wen, and Y. Li, "Instance segmentation using semi-supervised learning for fire recognition," *Heliyon*, vol. 8, no. 12, 2022, Art. no. e12375.
[29] Z. Guan, X. Miao, Y. Mu, Q. Sun, Q. Ye, and D. Gao, "Forest fire segmentation from aerial imagery data using an improved instance segmentation model," *Remote Sens.*, vol. 14, no. 13, p. 3159, Jul. 2022, doi: 10.3390/rs14133159.
[30] Y.-C. Zhou, Z.-Z. Hu, K.-X. Yan, and J.-R. Lin, "Deep learning-based instance segmentation for indoor fire load recognition," *IEEE Access*, vol. 9, pp. 148771–148782, 2021, doi: 10.1109/ACCESS.2021.3124831.
[31] C. Niu, H. Guo, and Y. Wang, "Fast flame recognition algorithm base on segmentation network," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces Abstr. Workshops (VRW)*, Shanghai, China, Mar. 2023, pp. 458–461, doi: 10.1109/VRW58643.2023.00099.
[32] L. Martins, F. Guede-Fernández, R. Valente de Almeida, H. Gamboa, and P. Vieira, "Real-time integration of segmentation techniques for reduction of false positive rates in fire plume detection systems during forest fires," *Remote Sens.*, vol. 14, no. 11, p. 2701, Jun. 2022, doi: 10.3390/rs14112701.
[33] H. Wei and Y. Huang, "Online multiple object tracking using spatial pyramid pooling hashing and image retrieval for autonomous driving," *Machines*, vol. 10, no. 8, p. 668, Aug. 2022, doi: 10.3390/machines10080668.
[34] D. Carmo, I. Campiotti, I. Fantini, L. Rodrigues, L. Rittner, and R. Lotufo, "Multitasking segmentation of lung and COVID-19 findings in CT scans using modified EfficientDet, UNet and MobileNetV3 models," in *Proc. 17th Int. Symp. Med. Inf. Process. Anal.*, vol. 12088, Dec. 2021, pp. 65–74.
[35] Y. Cai, Y. Long, Z. Han, M. Liu, Y. Zheng, W. Yang, and L. Chen, "Swin Unet3D: A three-dimensional medical image segmentation network combining vision transformer and convolution," *BMC Med. Informat. Decis. Making*, vol. 23, no. 1, p. 33, Feb. 2023.
[36] R. Shi, S. Yang, Y. Chen, R. Wang, M. Zhang, J. Lu, and Y. Cao, "CNN-transformer for visual-tactile fusion applied in road recognition of autonomous vehicles," *Pattern Recognit. Lett.*, vol. 166, pp. 200–208, Feb. 2023.
[37] Q. Wang, H. Fei, S. N. A. Nasher, X. Xia, and H. Li, "A macaque brain extraction model based on U-Net combined with residual structure," *Brain Sci.*, vol. 12, no. 2, p. 260, Feb. 2022, doi: 10.3390/brainsci12020260.
[38] S. Liu, B. Zhao, Y. Wang, M. Zhu, and H. Fu, "Squeeze-and-excitation blocks embedded YOLO model for fast target detection under poor imaging conditions," *Proc. SPIE*, vol. 12277, pp. 281–286, Jul. 2022.
[39] N. Zaghari, M. Fathy, S. M. Jameii, and M. Shahverdy, "The improvement in obstacle detection in autonomous vehicles using YOLO non-maximum suppression fuzzy algorithm," *J. Supercomput.*, vol. 77, no. 11, pp. 13421–13446, Nov. 2021.
[40] S. Ma, L. Wang, P. Chen, R. Qin, L. Hou, and B. Yan, "A mixed visual encoding model based on the larger-scale receptive field for human brain activity," *Brain Sci.*, vol. 12, no. 12, p. 1633, Nov. 2022, doi: 10.3390/brainsci12121633.
[41] A. Lou and M. Loew, "CFPNET: Channel-wise feature pyramid for real-time semantic segmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Anchorage, AK, USA, Sep. 2021, pp. 1894–1898, doi: 10.1109/ICIP42928.2021.9506485.
[42] X. Xu, Y. Jiang, and W. Chen, "DAMO-YOLO: A report on real-time object detection design," 2022, *arXiv:2211.15444*.
[43] Y. Li, H. Yang, J. Wang, C. Zhang, Z. Liu, and H. Chen, "An image fusion method based on special residual network and efficient channel attention," *Electronics*, vol. 11, no. 19, p. 3140, Sep. 2022, doi: 10.3390/electronics11193140.
[44] H. You, Y. Lu, and H. Tang, "Plant disease classification and adversarial attack using SimAM-EfficientNet and GP-MI-FGSM," *Sustainability*, vol. 15, no. 2, p. 1233, Jan. 2023, doi: 10.3390/su15021233.
[45] M. Gao, C. Chen, J. Shi, C. S. Lai, Y. Yang, and Z. Dong, "A multiscale recognition method for the optimization of traffic signs using GMM and category quality focal loss," *Sensors*, vol. 20, no. 17, p. 4850, Aug. 2020, doi: 10.3390/s20174850.
[46] Z. Leng, M. Tan, C. Liu, E. D. Cubuk, X. Shi, S. Cheng, and D. Anguelov, "PolyLoss: A polynomial expansion perspective of classification loss functions," 2022, *arXiv:2204.12511*.

XIANGHONG CAO was born in 1972. She is currently the Deputy Dean of the School of Architectural Environmental Engineering, Zhengzhou University of Light Industry, the Director of Henan Provincial Intelligent Building and Human Settlement Engineering Technology Research Center, a professor, and a master tutor. Her main research interests include building electrical and intelligent fire protection.

YIXUAN SU was born in 1999. He is currently pursuing the Graduate degree with the School of Architectural Environment Engineering, Zhengzhou University of Light Industry. His main research interests include fire detection, instance segmentation, YOLO algorithm, and image recognition.

XIN GENG was born in 1982. He is currently pursuing the Ph.D. degree. He is also a Lecturer with the School of Architectural Environmental Engineering, Zhengzhou University of Light Industry, and a master tutor. His main research interests include embedded system design and image recognition.

YONGDONG WANG was born in 1990. He is currently pursuing the Ph.D. degree. He is also a Lecturer with the School of Architectural Environment Engineering, Zhengzhou University of Light Industry, and a master tutor. His main research interest includes artificial intelligence spatio-temporal data analysis.