IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 March 2024, accepted 31 March 2024, date of publication 8 April 2024, date of current version 14 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3385856

THEORY

DCGC-YOLO: The Efficient Dual-Channel Bottleneck Structure YOLO Detection Algorithm for Fire Detection

YUN HE[1], JUNJIE HU[1], MING ZENG[1], YINGJING QIAN[2], AND RENMIN ZHANG[1,2], (Member, IEEE)

[1]School of Communication and Electronic Engineering, Jishou University, Jishou, Hunan 416000, China
[2]Key Laboratory of Intelligent Control Technology for Wuling-Mountain Ecological Agriculture in Hunan Province, Huaihua University, Huaihua, Hunan 418000, China

Corresponding authors: Yingjing Qian (Qyingjing@163.com) and Renmin Zhang (rzhang1981@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 62061017; in part by NSF of Hunan Province under Grant 2021JJ30556, Grant 2021JJ30545, and Grant 2022JJ50310; in part by the Education Commission of Hunan Province under Grant 22B0761 and Grant 2020JGZD064; in part the Graduate Research and Innovation Project of Hunan Province (Enterprise Collaborative Project) under Grant QL20230254; and in part by Huaihua University under Grant ZNKZD2021-02.

ABSTRACT Fast and highly accurate fire detection algorithms are crucial for production and daily life. However, detecting fires in the early stages is challenging due to the lack of distinct features and fixed shapes of flames and smoke. To address this issue, we propose a fire detection algorithm, DCGC(Dual Channel Group Convolution)-YOLO, based on an improved YOLOv5 model. The DCGC-YOLO model introduces a new layer structure and anchor algorithm to optimize original YOLOv5 model. Firstly, we introduce a Cross Stage Partial (CSP) structure with a cascade of large convolutional kernels in the bottleneck layer. This structure increases network's receptive field, enhances feature extraction capabilities, and employs a channel cleansing mechanism that combines various channels separated by group convolution, promoting information exchange in the channel dimension and enabling better information encoding. Next, we integrate the Effective Squeeze and Extraction (eSE) mechanism into the new layer structure, enhancing the model's ability for long-range modeling and focusing more on target areas. Finally, we utilize an Intersection over Union (IoU)-based anchor generation algorithm to adjust the anchor sizes on custom fire dataset, enhancing the model's robustness and improving detection accuracy. Experimental results on our custom fire dataset demonstrate that the proposed DCGC-YOLO algorithm effectively detects targets with mAP of 41.1%, which is 2.9% higher than YOLOv5s, while reducing network parameters and computational complexity. To further validate the effectiveness of our proposed algorithm, we conduct experiments on the COCO2017 dataset. The results show that DCGC-YOLO achieves mAP of 38.9%, demonstrating strong generalization and competitiveness compared to state-of-the-art detectors.

INDEX TERMS Computer vision, object detection, fire detection, smoke detection, YOLOv5.

I. INTRODUCTION

Fire occurs frequently and in various locations, making it being one of the most common and widespread disasters that threaten public safety and social development. According to a report from the National Fire Protection Association in the United States [1], in 2021, there were approximately 1.35 million fires in the country, resulting in 3,800 deaths, 14,700 injuries, and property losses totaling $15.9 billion. Basically, fire detection methodology can be roughly organized the following two categories: one is based on sensor detection [2], [3], [4], and the other is based on vision detection. Traditional sensor detection mainly includes smoke sensors, heat sensors, and gas sensors. They sense changes in the surrounding environment based on smoke, temperature, and heat in the air. When these conditions

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

reach a certain threshold, these sensors will trigger fire alarms. Due to the limitations of the working principles of traditional sensors, they cannot detect fires in the early stages and have a high rate of false alarms. In addition, they cannot determine the location of the fire source, and require regular maintenance and replacement, which increases costs. With the development of deep learning technology and GPU computing power, the visual-based fire detection technology has been rapidly advancing.

Generally, vision-based fire detection techniques can be seen as a specialized form of object detection task. It captures videos or images through cameras or other image acquisition devices, and then uses algorithms for image processing and analysis to achieve fire detection. Compared with traditional sensor-based methods, visual-based fire detection technology has the following advantages: High sensitivity, Visualization, High precision, Low cost, High flexibility. Raj and Prabadevi [5], [6]. improved YOLOv7 and improved the detection accuracy of Steel Tube and Steel Strip. These characteristics have attracted researchers, and a series of flame and smoke recognition models have been developed through research. However, there are many existing strategies that pursue detection speed but result in a decrease in accuracy. Balancing detection speed and accuracy is an urgent problem that needs to be addressed. Many visual-based fire detection models have been created by scholars, there are still many problems, such as background environment interference, changes in lighting intensity, diversity of flame and smoke shapes, and a lack of large-scale real fire scene data. These issues can all affect the detection results.

To address these issues, this paper proposes a lightweight real-time detection algorithm suitable for fire detection. The algorithm performs well not only on fire datasets but also on general object detection tasks. We redesigned the network structure based on the YOLOv5 network, taking inspiration from ConvNeXt [7] and ShuffleNetV2 [8]. We designed a large convolutional kernel cascaded dual-channel group convolution residual structure to effectively capture features for targets with significant scale variations like fire and smoke, thus reducing false positives. To further distinguish fine-grained features between targets and backgrounds, we proposed a feature optimization module, which includes the effective Squeeze and Extraction Block (eSEB) [9] and Channel Shuffle module. The eSEB explicitly models the inter-channel dependencies in feature maps to enhance feature representation. The Channel Shuffle module enhances information exchange among groups of feature maps, improving the model's generalization capability. Additionally, we re-clustered the anchor boxes of the dataset using the IoU K-means clustering algorithm and accelerated model convergence by using the CIoU [10] loss function. Finally, we constructed a fire dataset that includes two categories: fire and smoke.

In summary, the main contributions of this paper are as follows:

1) We design a structure that captures target features using a large convolutional kernel cascaded structure, taking into account the wide range of target size distributions in the dataset. Group convolution is employed to reduce computational complexity.
2) We address the lack of long-range modeling capability and weakened inter-group information flow in the aforementioned structure by introducing the eSEB and Channel Shuffle module. These components effectively enhance the detection capability and robustness of the model.
3) We propose an IoU-based K-means algorithm that replaces Euclidean distance with IoU to measure the overlap between anchor boxes and annotated boxes, alleviating the issue of wide target size distribution in the dataset.

The following section are arranged as follows: Section II introduces the fire dataset and related work on fire recognition and datasets in recent years. Section III provides a detailed description of the proposed algorithm. Section IV conducts ablation experiments on different algorithms, and analyzes the experimental results. Finally, the main content and future research directions of this study are proposed.

II. RELATED WORK

A. TRADITIONAL METHODS FOR FIRE DETECTION

Traditional fire detection methods can be reviewed from two aspects: literature on sensor devices and literature on fire image processing techniques. The main types of fire detection sensor technologies are Heat sensing [11], Gas Sensing [12], Flame Sensing [13], Smoke Sensing [3], and Miscellaneous Sensing [14]. Fire detectors based on heat sensing are typically suitable for indoor environments with low false alarm rates, but they may not be able to detect fires in the early stages. Gas sensors detect gases such as CO_2, CO, and HCN that are produced during a fire. They have high sensitivity but require stable operation in high-temperature environments, which increases their cost. Flame sensors rely on the visible light and infrared radiation emitted by flames to detect the occurrence of a fire. However, they have distance limitations and can be affected by infrared obstruction and thermal reflection. Smoke sensors detect the level of smoke particles in the air but have a higher false alarm rate. They are generally used in combination with other detection methods. Traditional fire detection methods struggle to achieve a good balance between detection cost and accuracy.

Fire detection systems based on sensors are often limited by the sensors themselves, leading to high false alarm rates or detection distance limitations [15]. In contrast, methods based on image processing demonstrate better accuracy. Traditional visual-based fire detection methods typically involve the following three steps: 1) fire area identification, 2) extraction of fire or smoke features, and 3) classification using machine learning techniques. For the first step, several methods are commonly employed, including: For static image detection, the main approach is based on color methods, including based on color spaces [16], [17], HSV [18], L*a*b [19], YUV [20], YCbCr [17], [21] and LUV [22], and the method based on grayscale image [23]. These methods utilize the distinct color of flames in different color spaces to identify fire. However, they are susceptible to lighting variations and can be affected by objects that resemble fire, which are their limitations.

B. CNN-BASED METHODS FOR FIRE DETECTION

The convolutional neural networks(CNN)-based methods have become increasingly popular in object detection and image classification. Compared to traditional detection methods, CNN-based approaches require less manual intervention and exhibit strong feature extraction capabilities and good generalization. In the past few years, many researchers have developed fire detection models based on CNN architectures. Methods based on visual detection are generally divided into two categories. The first is the one-stage detection model, such as the You Only Look Once (YOLO) series [24], Single Shot MultiBox Detector (SSD) [25], and RetinaNet [26]. The second is the two-stage detection model, represented by models such as Faster R-CNN [27], Cascade R-CNN [28], and Mask R-CNN [29]. Xue et al. [30]. improved the YOLOv5 [31] algorithm by introducing the CBAM [32] attention mechanism and adopting BiFPN [33], which makes the network more focused on fire information. Xu et al. [34]. used EfficientNet to guide YOLOv5 for fire recognition to minimize false positives without adding extra latency. Muhammad et al. proposed a lightweight and computationally efficient CNN model using the SqueezeNet architecture [35]. This model achieves a balance between fire detection speed and accuracy. It has been validated for deployment in closed circuit television (CCTV) surveillance networks. Zhao et al. proposed an improved YOLOv4 model by using EfficientNet as the backbone network and adding a small object detection layer [36]. The results showed that it outperformed Faster R-CNN and YOLOv3 in terms of accuracy on their custom fire dataset. Jandhyala et al. employed the CNN-based InceptionV3 model for classifying aerial images and used the Single Shot Detector model to detect smoke regions [37]. The model was pretrained on the COCO dataset and fine-tuned on the aerial dataset. It achieved a classification accuracy of 88% and a detection accuracy of 91% on the aerial dataset. Wang et al. designed a more efficient layer aggregation-based video fire detection network to detect the brightness and chromaticity of flames. This network improved the detection accuracy by 3.5% on public datasets and 2.3% on their custom dataset [13].

C. FIRE DATASETS

A dataset is one of the foundations for deep learning models to recognize patterns. Currently, there is no authoritative, fully open and deep learning-compatible dataset available for fire recognition in both academia and industry. Therefore, constructing a fire dataset is a key element for fire recognition. So far, some of the mainstream fire datasets that have been partially open-sourced mainly contain AVI-format video data or low-resolution images. They include smoke images with various background interference factors such as different fire-like objects, long shots, close-ups, night scenes, and more. The remaining self-built datasets are mostly labeled with a single category, a single environment, or lack sufficient data, resulting in limited scene diversity. These limitations greatly affect the ability of deep learning algorithms to recognize flames and smoke, leading to decreased accuracy, poor generalization, and insufficient robustness. Therefore, constructing a large-scale, multi-scene fire dataset is of great significance for establishing an efficient fire safety system. Some of the major fire datasets are shown in Table 1.

Due to the limited availability of open-source datasets for fires, most existing datasets lack annotations, particularly in video data, making them suitable only for detection purposes. Therefore, we have constructured a fire dataset. The dataset for this experiment is composed of images from public datasets like ImageNet, VisFire, BoWFire, as well as flame and smoke images obtained through online searches. The dataset encompasses scenes such as forest fires, indoor fires, factory fires, vehicle fires, building fires, outdoor fires, drone aerial shots, and candle flames. As shown in Table 2, the dataset consists of 28,260 images, including 22,407 fire images, 17,073 smoke images, 36,463 fire bounding boxes, and 20,975 smoke bounding boxes. The training set comprises 16,960 images, while the validation and test sets each include 5,650 images. Given the non-rigid nature of flames and smoke, lacking fixed shapes and being susceptible to human factors during annotation, the experiment employs maximum bounding rectangle annotation to minimize background information for the targets. Thin edges of fire and smoke are considered to a certain extent during annotation.

III. METHODOLOGY

A. YOLOV5 OVERALL REVIEW

YOLOv5 is an excellent target detector with a good balance between speed and accuracy. YOLOv5 has five different versions, YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, which are implemented by adjusting the width and depth of the network. The backbone of YOLOv5 uses CSPDarkNet53 and SPPF structure, and the FPN uses PANet [57] and uses a 1×1 convolutional layer as the prediction head of YOLOv5. Different network models are implemented by adjusting the depth and width of the network.

B. THE FRAMEWORK OF DCGC-YOLO

We proposed the DCGCLayer module and modified the original YOLOv5 specifically for fire detection. The overall framework of DCGC-YOLO is shown in Figure 1. First, the input image size is scaled to the interval [320, 960] (an integer multiple of 32), and then, the input is fed to

TABLE 1. Comparison of different fire datasets.

Labeling Type	Dataset Name	Describe	Task	Perspective	Year
Smoke	Research Webpage about Smoke Detection for Fire Alarm: Datasets [38]	3 smoke videos 3 non smoke videos 6323 smoke image 74,989 non smoke images	Detect	Terrestrial	2017
	State Key Laboratory of Fire Science Dataset	3578 smoke images 30,000 synthetic smoke image	Detect	Terrestrial	-
Fire	Corsican Fire Dataset [39]	500 fire and smoke images	Detect Segmentation	Terrestrial Aerial	updating
	BoWFire Dataset [40]	Train: 80 fire images 160 no fire images Test: 119 fire images 107 no fire images	Detect Segmentation	Terrestrial	2015
	CAIR's Fire Detection Image Dataset [41]	651 images	Detect	Terrestrial	2017
	FireNet [42]	502 images	Detect	Terrestrial	2019
	The FLAME dataset [43]	47,992 frames	Detect Segmentation	Aerial	2020
	Dataset for Forest Fire Detection [44]	1900 images	Detect	Terrestrial	2020
	Fire Detection by Dhruvil Shah [45]	3225 images	Detect	Urban	2020
	LANDSAT-8 [46]	31,000+ images	Detect Segmentation	Aerial	2020
	Harbin Institute of Technology Flame Dataset	82,443 fire images 17,312 no fire images	Detect	Terrestrial	2022
Mixed	Türkiye Bilkent University Flame Video Library	38 video	Detect	Terrestrial	2006
	KMU Fire & Smoke Database [47]	1 fire video 2 smoke video 1 smoke or flame-like moving object	Detect	Terrestrial	2012
	MIVIA Fire Detection Dataset [48]	14 fire video 17 no fire video 149 smoke video	Detect	Terrestrial Aerial	2015
	Furg Fire Dataset [49], [50]	365,702 frames	Detect	Urban	2018
	Fire Detection From closed-circuit television [51]	864 frames	Detect	Terrestrial	2019
	AIDER [52], [53]	740 images	Detect	Aerial	2020
	DataCluster Labs' Fire and Smoke Dataset [54]	7000+ frames	Detect	Terrestrial	2020
	Drone-Collected RGB/IR Image Dataset[46]	53,451 frames 9 fire and smoke video(no label)	Detect	Terrestrial Aerial	2022
	FLAME 2 [55]	53,451 frames	Detect	Aerial	2022
	FASDD [56]	82,666 fire object 62,404 smoke object	Detect	Terrestrial Aerial	2023

TABLE 2. Experimental dataset details.

Dataset	Fire Image	Smoke Image	Fire Annotation	Smoke Annotation
Train	13464	10162	21786	12508
Val	4434	3480	7197	4247
Test	4509	3431	7480	4220
Total	22407	17073	36463	20975

the feature extraction network with DCGCBlock module, which increases the perceptual field of the network. Then, the three feature maps output from the feature extraction network are input to the feature fusion network, which further fuses different dimensional features and outputs three feature vectors rich in semantic information. Finally, the CIoU loss function is used for back propagation to speed up the model convergence and improve the robustness of the model.

C. IMPROVE THE STRUCTURE OF DCGCLAYER

1) DCGCLAYER

The DCGCLayer, proposed by us, addresses these issues by replacing the bottleneck module of the original network with the DCGCBlock module, which concatenates smaller convolutional kernels before larger ones. This modification increases the network's receptive field. The YOLOv5 network, enhanced with the DCGCLayer, is capable of

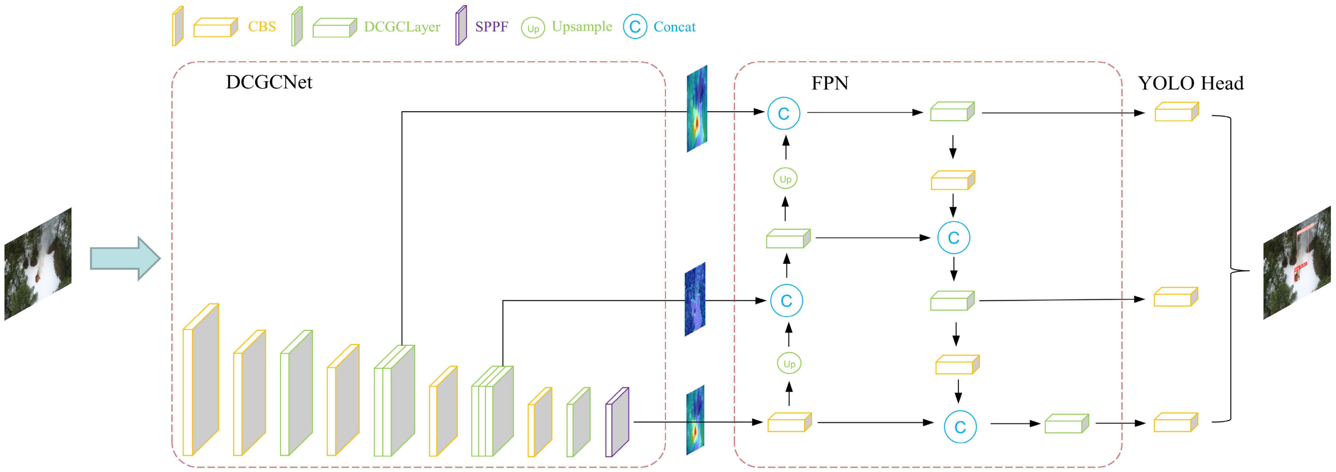

FIGURE 1. The overall framework of the DCGC-YOLO network is based on YOLOv5, with the proposed DCGCLayer replacing the C3 layer. CSB = Conv + BN + SiLU, the proposed DCGCLayer is introduced in Section III-C. The YOLO head consists of 1 × 1 convolutional layers.

TABLE 3. Comparison between ordinary convolution and group convolution.

	Input Size	#Params(M)	FLOPs(G)
DCGCLayer	20×20×512	1.082	0.329
C3	20×20×512	1.182	0.474

effectively recognizing smoke and flames with significant scale variations. Compared to the YOLOv5's C3 module, the DCGCLayer exhibits a reduction of 30.6% and 8.4% in computational and parameter complexity, respectively, as shown in Table 3.

The schematic diagram of the DCGCLayer is shown in Figure 2(b). Building upon the YOLOv5's C3 structure, we replaced the stacked Block structure and introduced an additional eSE attention mechanism layer. The structure of DCGCBlock is depicted in Figure 3(d), employing a dual-branch CSP design. Branch one consists of a concatenation of convolutional BN layers with kernel sizes of 9×9, 3×3, and 1×1 (forming a CBS layer). Branch two is composed of convolutional BN layers with kernel sizes of 3×3 and 1×1. Both the 9×9 and 3×3 convolutional layers employ group convolution. This architecture significantly enhances the network's receptive field, enabling it to adapt to objects with substantial size variations. Subsequently, the two branches are concatenated along the channel dimension. We avoided the use of addition here because element-wise addition of feature maps would significantly decrease the network's computational speed. Next, the channel shuffle mechanism is applied to reorganize the feature maps separated by group convolution, enhancing information flow. Finally, the output passes through a CBS layer and a residual layer.

2) ESE ATTENTION

In object detection, the attention mechanism is used to improve the accuracy and efficiency of the model by selectively focusing on the most relevant parts of an image for object detection.

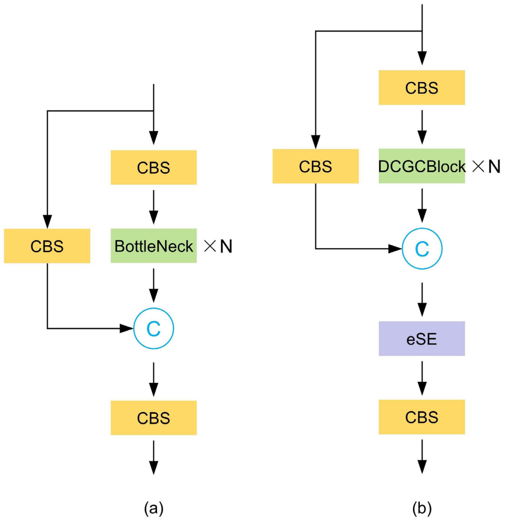

FIGURE 2. (a) C3 model (b) DCGCLayer model.

There are different types of attention mechanisms used in object detection, but one of the most common is spatial attention, which focuses on specific regions of the image. Spatial attention can be used in combination with CNNs, which are commonly used for object detection, to identify the most important regions of an image for object detection. The attention mechanism is used to weight the importance of different regions of the image based on their relevance for object detection. Overall, the attention mechanism in object detection helps to improve the accuracy and efficiency of the model by selectively focusing on the most relevant parts of the image for object detection, rather than processing the entire image indiscriminately.

eSE attention mechanism refers to "Effective Squeeze and Extraction" which is a type of attention mechanism proposed in a research paper by authors Jianping He, et al.

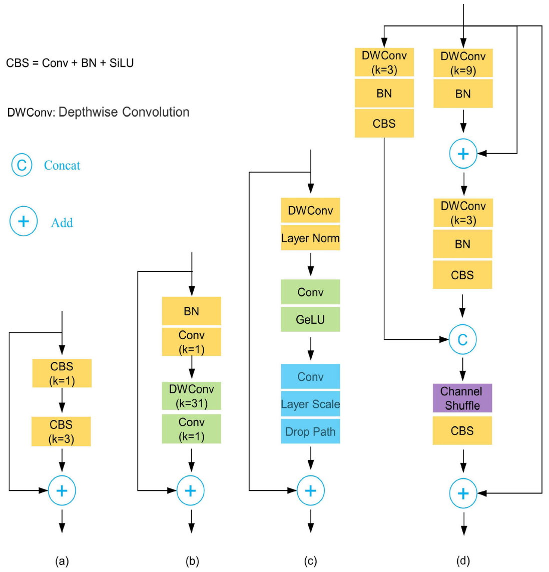

FIGURE 3. (a) BottenNeck (b) RepLK Block (c) ConvNeXt Block (d) DCGCBlock. The unannotated branches in the diagram are all identity mappings.

The eSE attention mechanism is designed to improve the performance of deep CNNs by selectively emphasizing informative channels in the intermediate feature maps of the network.

In the eSE attention mechanism, a set of convolutional layers are used to learn the channel-wise dependencies between the feature maps of a CNN. The learned dependencies are then used to compute a set of channel attention weights that selectively emphasize informative channels while suppressing uninformative ones. The channel attention weights are multiplied with the feature maps to generate the final output feature maps.

Compared to other attention mechanisms, eSE attention is computationally efficient and can be easily integrated into existing CNN architectures without adding significant computational overhead. It has been shown to improve the performance of various image recognition tasks, including object detection, image classification, and semantic segmentation.

The principle of the eSE attention mechanism is shown in Figure 4. The input image is first fed into an average pooling layer - F_{avg}, and then into a 1 × 1 convolutional layer - W_c (unlike the SE attention mechanism, there is no compression of the channel dimension, allowing for the preservation of as much feature map information as possible). Finally, the output feature map with different weights is obtained through a h-sigmoid activation function.

The calculation method for the eSE attention mechanism is as follows:

$$X_{out} = X_{in} A_{eSE} \left(X_{in}\right) \tag{1}$$

$$A_{eSE} = \sigma \left(W_c \left(F_{avg} \left(X_{in}\right)\right)\right) \tag{2}$$

$$h - sigmoid(x) = max\left(0,\ min\left(1,\ \frac{2x+5}{10}\right)\right) \tag{3}$$

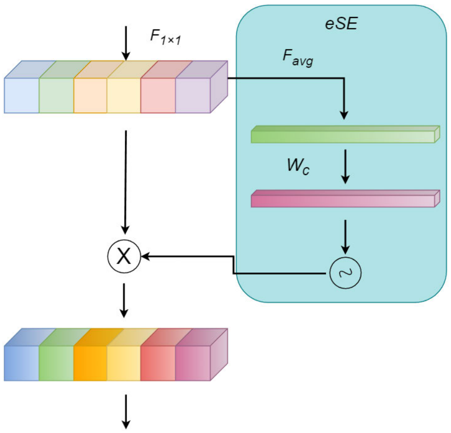

FIGURE 4. eSE attention principle.

In Formula (1), X_{in} and Xout represent the input feature map and the output feature map, respectively, and A_{eSE} represents the eSE attention operation. In Formula (2), X_{in} represents the input feature map, F_{avg} is the average pooling layer applied to each channel of the feature map, W_c is a 1×1 convolutional layer with the number of convolutional kernels being C/r (where C is the input dimension, and in this paper, r=1). σ represents the h-sigmoid activation function, and its calculation method is shown in Formula (3).

3) VISUAL RESULTS

To validate the effectiveness of the proposed backbone, we visualized the 4th, 6th, and 9th layers of both DCGCNet and CSPDarkNet53, as these three layers serve as input feature layers for the FPN. Figure 5 shows the visualized heatmap results of feature extraction. The results indicate that DCGCNet can extract more effective features. Specifically, in the 4th and 6th layers, DCGCLayer can preserve more low-level texture information. Moreover, in the overall output feature map of the backbone (9th layer), DCGCNet can generate more important features, whereas CSPDarkNet53 neglects some valuable information.

D. IOU KMEANS

Predefined anchors are crucial for the YOLOv5 detection algorithm. The default anchors used by the YOLOv5 algorithm are based on the COCO2017 dataset, which is suitable for general-purpose object detection. However, in this study, we used a fire dataset where the aspect ratios of the target sizes vary significantly. Using the default anchors would lead to reduced detection accuracy. Therefore, to improve the detection accuracy of flames and smoke, we employed the IoU kmeans algorithm to calculate the anchors specific to the fire dataset.

Predefined anchors are crucial for the YOLOv5 detection algorithm. The default anchors used by the YOLOv5 algorithm are based on the COCO2017 dataset, which is suitable for general-purpose object detection. However,

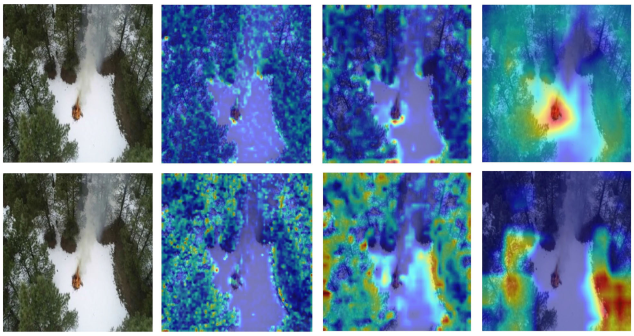

FIGURE 5. The feature maps extracted from DCGCNet (top) and CSPDarkNet53 (bottom) are visualized. From left to right, the visualizations show the original image, Feature4, Feature6, and Feature9 heatmap(The feature map numbers are determined based on the Layer Number specified in Table 4), respectively. In the lower-level feature maps, DCGCNet can retain more spatial information, while in the higher-level feature maps, it can extract more semantic information.

TABLE 4. Anchor box size.

Feature map	IoU Kmeans anchor box size
80×80	(14,17) (30,32) (75,42)
40×40	(45,75) (103,92) (223,112)
20×20	(143,201) (323,222) (500,415)

Algorithm 1 IoU Kmeans Algorithm for Anchor Box Size

```
Input:
    $B = B_1 \ldots, B_N, K, n$
    $B$ is a numpy array containing the width and height of the bounding boxes in the dataset
    $N$ is the total number of bounding boxes in the dataset
    $K$ is the number of clusters in the k-means algorithm
    $n$ is the maximum number of iterations for the for loop
Output:
    $C_k$
Initialization:$C_k \in C_N$ and $LC = 0$, $count = 0$
while True do
    for $B_i \in B_N$ do
        for $C_j \in C_N$ do
            $D_i = 1 - \text{IoU}(B_i, C_j)$
        end for
        $NC \leftarrow \min(D_{j,k})$
        Assign each annotated box to the nearest cluster center based on "distance"
        if $(LC) == (NC)$ then
            break
        end if
    end for
    for $C_j \in C_k$ do
        $C_j \leftarrow median(Box[NC = k])$
        Update the cluster centers by using the median of each class of boxes in the dataset as the new cluster centers
    end for
    $LC \leftarrow NC$
    count++
    if $count > n$ then
        break
    end if
end while
```

in this study, we used a fire dataset where the aspect ratios of the target sizes vary significantly. Using the default anchors would lead to reduced detection accuracy. Therefore, to improve the detection accuracy of flames and smoke, we employed the IoU kmeans algorithm to calculate the anchors specific to the fire dataset. The k-means algorithm is a simple and effective method. When computing anchor sizes for a dataset using the k-means algorithm, the commonly used metric is the width and height of the target bounding boxes (i.e., Euclidean distance), which groups boxes with similar dimensions into the same cluster. However, this approach may lead to larger errors in the clusters of larger boxes compared to smaller boxes. Therefore, using IoU as the box clustering metric is more reasonable. In this study, we use 1-IoU as the metric, transforming the optimization problem from maximizing to minimizing. Since the YOLOv5 algorithm uses three feature detection layers and each prediction grid uses three anchors, we have nine clustering centers for this study. For the shallow feature maps, larger anchor sizes are used, while for the deep feature maps, smaller anchor sizes are used. The target boxes from the fire dataset are used for clustering, and the anchor sizes are shown in Table 4.

IV. EXPERIMENTS

A. TRAINING DETAILS AND RESULTS

1) TRAINING DETAILS

DCGC-YOLO was pretrained for 300 epochs on the COCO2017 dataset using the YOLOv5s pretrained weights. It was then fine-tuned for 300 epochs on the fire dataset. The initial learning rate was set to 0.01, and cosine annealing was used for learning rate decay. The SGD optimizer was used (AdamW optimizer was used on the fire dataset). The network was implemented using the PyTorch framework. The experiments were conducted on a Windows 11 platform environment with an i9-13900k@3GHz CPU, 32.0GB of memory, CUDA version 11.1, CUDNN version 8.0.5, Pytorch version 1.9.0, GeForce NVIDIA GTX4090 24G GPU model, and Python version 3.8.16.

B. COMPARISONS OF THE PROPOSED METHOD WITH SOME STATE-OF-THE-ART ALGORITHMS ON THE FIRE DATASET

We compared the performance of our proposed DCGC-YOLO with several state-of-the-art methods on the fire dataset. We evaluated the performance of these models using various AP metrics, Model Parameter Count (Params), and Floating Point Operations Per second (FLOPs). Table 5 presents the test results of all the involved models, where

TABLE 5. Performance results of OUR METHOD with advanced algorithms on the fire dataset.

Method	Backbone	Input Size	AP(%)	AP_{fire}	AP_{smoke}	AP^{val}_{50}	AP^{val}_{75}	AP^{val}_{s}	AP^{val}_{m}	AP^{val}_{l}
Faster R-CNN	ResNet50	608	38.2	37.0	39.4	70.1	37.1	15.8	26.7	47.3
Mask R-CNN	ConvNeXt-Tiny	640	36.1	35.1	37.0	67.9	34.8	11.6	25.7	45.3
SSD	VGG16	512	34.5	33.9	35.1	66.4	32.1	12.7	24.9	42.6
RetinaNet	ResNet101	640	37.9	36.3	39.5	69.2	36.6	15.5	26.6	47.4
FCOS	ResNet50	640	36.9	35.6	38.1	68.5	35.6	15.6	25.8	46.2
YOLOv3-Tiny	DarkNet53	640	25.4	31.0	19.8	57.4	18.7	11.9	21.7	30.7
YOLOv3	DarkNet53	608	37.1	37.6	36.6	68.1	35.3	15.8	25.7	45.7
YOLOv4-Tiny	CSPDarkNet53	640	28.3	33.5	23.1	61.4	22.0	13.9	24.3	33.4
YOLOv5s	Modified CSP v5	640	38.2	38.9	37.5	70.1	36.4	15.8	26.8	46.7
YOLOv7-Tiny-SiLU	-	640	39.1	39.7	38.6	70.7	38.1	15.7	27.2	47.9
YOLOX-Tiny	Modified CSP v5	640	38.8	38.8	38.8	70.4	38.3	16.0	27.9	47.2
YOLOv8n	-	640	39.8	39.9	39.7	71.2	38.6	15.6	28.2	48.3
i Proposed framework	DCGCNet	640	41.1	40.8	41.5	73.1	40.6	16.3	29.8	50.2

the metrics represent the best performance achieved by each model. It is worth noting that Faster R-CNN [27], Mask R-CNN [28], SSD [25], FCOS [58], RetinaNet [26], and YOLOX [59] are implemented based on mmdetection3.0.0 [60]. Their respective backbone networks are shown in Table 6. The backbone network of Mask R-CNN is based on ConvNext-Tiny. To validate the effectiveness of DCGC-YOLO detection, we compared it with other state-of-the-art methods using different AP metrics, and DCGC-YOLO achieved real-time high-precision detection compared to these methods.

1) DETECTION PERFORMANCE

Due to the difficulty in defining the boundaries of smoke and flames themselves, and our annotation method using maximum bounding boxes, the detection accuracy may not appear to be very high, even though the dataset is not large. We use the classic evaluation metric AP in the field of object detection to validate the effectiveness of the method, with Mask R-CNN using the bbox evaluation metric. As shown in Table 5, DCGC-YOLO achieves the highest AP in terms of average detection precision, with a value of 40.5%. It outperforms the second-highest YOLOv7-Tiny [61] by 1.4%, and the third-highest YOLOX-Tiny by 1.7%. Compared to the baseline (YOLOv5s), our AP is 2.3% higher. The two-stage algorithms, Faster R-CNN and Mask R-CNN, only achieve 38.2% and 36.1% respectively. Compared to the state-of-the-art algorithm YOLOv8n, the proposed algorithm achieves a 1.3% higher mAP.

The YOLOv5s algorithm achieves an AP of 38.9% for flame detection and 37.5% for smoke detection. It outperforms most advanced methods in flame detection but falls behind other methods in smoke detection. This may be attributed to the strong generalization performance of the YOLO series framework but relatively weaker feature extraction capability of the C3 module when it comes to smoke targets that are similar to the background.

We introduce the DCGCLayer module to improve YOLOv5s, and the results show that this module has a stronger ability to distinguish foreground and background, making it well-suited for fire detection. In terms of detecting large, medium, and small objects, DCGC-YOLO also outperforms other advanced methods, achieving the highest AP values.

2) FLOPS AND PARAMETERS

FLOPs and Parameters are important metrics that measure the size and computational complexity of deep learning models. In this paper, we chose YOLOv5s as the baseline model, which is a lightweight model. It is of great significance for subsequent model research and application deployment. Therefore, we compared the FLOPs and Parameters of existing methods and our proposed method, as shown in Table 6. The experimental results show that our proposed DCGC-YOLO model has lower FLOPs and Parameters than YOLOv5s. In terms of parameter count, it is higher than YOLOv4-Tiny, YOLOv7-Tiny, and YOLOX-Tiny, YOLOv8n. In terms of FLOPs, it is higher than YOLOv3-Tiny [62], YOLOv7-Tiny, and YOLOX-Tiny. In the lightweight model category, our proposed method can maintain detection accuracy while reducing model parameters and computational complexity. This is because we use large convolutional kernels and group convolution, which can effectively increase the receptive field of the network, extract deeper semantic information, and reduce model complexity.

3) VISUAL RESULTS

In this section, we randomly selected 3 images from the validation set for detection, and the results are shown in Figure 6. In the detection of smoke, the original YOLOv5s network exhibits weaknesses. When the background color is similar to the color of smoke, the YOLOv5s network faces issues of false negatives, which is particularly crucial for early warning of fires. In the detection of flames, the improved YOLOv5 network shows higher accuracy in detection, more precise target localization, and enhanced capability in extracting target texture features. This is

TABLE 6. Computational complexity and parameter of different methods.

Method	Backbone	Input Size	Params(M)	FLOPs(G)
Faster R-CNN	ResNet50	608	41.3	37.0
Mask R-CNN	ConvNeXt-Tiny	640	47.6	102
SSD	VGG16	512	24.5	87.6
RetinaNet	ResNet101	640	36.3	38.8
FCOS	ResNet50	640	32.1	37.3
YOLOv3-Tiny	DarkNet53	640	8.6	12.9
YOLOv3	DarkNet53	608	61.5	154.6
YOLOv4-Tiny	CSPDarkNet53	640	6.1	14.5
YOLOv5s	Modified CSP v5	640	7.0	15.8
YOLOv7-Tiny-SiLU	-	640	6.0	13.0
YOLOX-Tiny	Modified CSP v5	640	5.0	7.5
YOLOv8n	-	640	3.0	8.1
Proposed framework	DCGCNet	640	6.5	13.9

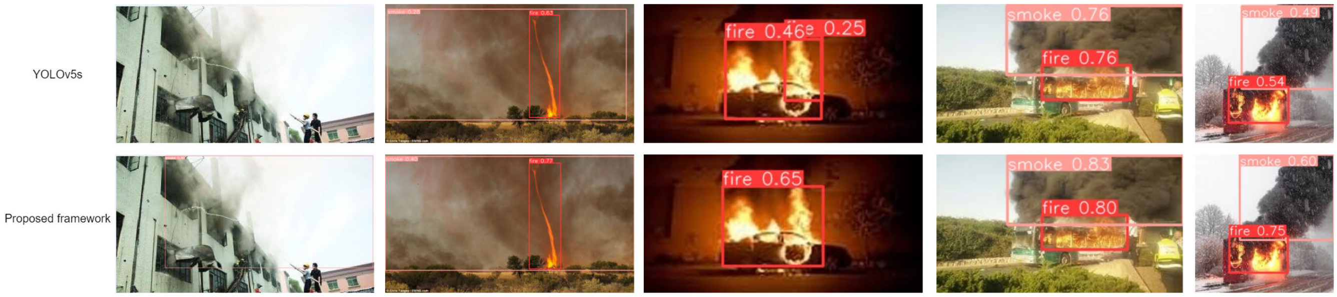

FIGURE 6. Visual detection results of various frameworks for the fire dataset.

of significant importance for achieving high-precision fire detection. Overall, the YOLOv5 network improved in this study demonstrates better adaptability to the requirements of fire detection.

C. ABLATION STUDY ON DCGC-YOLO

We conducted a burning experiment on the fire dataset to validate the effectiveness of the proposed module, and the results are presented in Table 7. In models (a), (b), and (c), we tested the IoU K-means, eSE attention, and DCGCBlock, respectively. The baseline model was based on YOLOv5s, and the evaluation metrics used were mAP at IoU thresholds of 0.5:0.95 and 0.5. The experimental results are shown in Table 7, In Model (a), we used the IoU K-means algorithm to calculate the anchor sizes for the fire dataset, resulting in an improvement of 0.8% in mAP for both thresholds. In Model (b), we introduced the eSE attention mechanism into the C3 module of YOLOv5s. This led to an increase of 0.9% and 1.4% in mAP for the respective thresholds. However, the parameter count increased by 0.7M, and the computational complexity increased by 0.8 GFLOPs. In Model (c), we replaced the Bottleneck module in the C3 module with the DCGCBlock module. This resulted in an improvement of 2.2% and 2.5% in mAP for the respective thresholds. Moreover, the parameter count decreased by 0.5M, and the computational complexity decreased by 2 GFLOPs. This was achieved by the DCGCBlock's large convolutional kernel in series mechanism, which greatly increased the receptive field of the network. The introduction of group convolution and the bottleneck-like structure increased the number of channels without increasing FLOPs. Additionally, channel shuffle operations were employed to enable information exchange between different channels. These enhancements made the proposed DCGCBlock module more powerful in feature extraction compared to the C3 module.

In hybrid models (d) and (e), we tested different strategy models, and the mAP of model (d) increased by 1.5% and 1.7% respectively, while the mAP of model (e) increased by 2.3% and 2.8% respectively. Overall, compared to the baseline, the parameter count reduced by 0.5M, the computational complexity reduced by 2.0GFLOPs, and the mAP increased by 2.8% and 3.3% respectively. Therefore, the proposed method can effectively improve the detection performance.

D. RESULTS ON THE MS COCO2017 DATASET

To further evaluate the generalization ability of the DCGC-YOLO, we conducted testing on the MS COCO 2017 dataset. Unlike the fire dataset, COCO 2017 consists of 80 different object categories and is one of the most widely used publicly available datasets for object detection. Table 8 presents the testing results of DCGC-YOLO and other SOTA models.

TABLE 7. Results of Ablation experiments.

model	IoU K-means	eSE attention	DCGCBlock	AP(%)	AP_{50}^{val}	#Params(M)	FLOPs(G)
YOLOv5s				38.3	69.8	7.0	15.8
(a)	✓			39.1	70.6	7.0	15.8
(b)		✓		39.2	71.2	7.7	16.6
(c)			✓	40.5	72.3	6.5	13.8
(d)	✓	✓		39.8	71.5	7.7	16.4
(e)		✓	✓	40.6	72.6	6.5	13.8
Proposed framework	✓	✓	✓	41.1	73.1	6.5	13.8

TABLE 8. Performance comparison results in COCO 2017.

Method	Backbone	Input Size	Params(M)	FLOPs(G)	AP(%)	AP_{50}^{val}
Faster R-CNN	ResNet50	-	41.3	37.0	37.4	58.1
Mask R-CNN	ConvNeXt-Tiny	-	47.6	102	46.2	68.1
SSD	VGG16	512	24.5	87.6	29.5	49.3
RetinaNet	ResNet101	-	36.3	38.8	36.5	55.4
FCOS	ResNet50	-	32.1	37.3	36.6	56.0
YOLOv3-Tiny	DarkNet53	416	-	5.5	-	33.1
YOLOv3	DarkNet53	608	61.5	154.6	33.0	57.9
YOLOv4-Tiny	CSPDarkNet53	416	6.1	6.9	24.9	40.2
YOLOv5s	Modified CSP v5	640	7.2	16.5	37.4	56.8
YOLOv7-Tiny-SiLU	-	640	6.2	13.8	38.7	56.7
YOLOX-Tiny	Modified CSP v5	416	5.0	6.4	32.8	-
YOLOv8n	-	640	3.2	8.7	37.3	-
Proposed framework	DCGCNet	640	6.7	14.5	38.9	58.4

DCGC-YOLO achieves an mAP 0.5:0.95 of 38.9% and an mAP 0.5 of 58.4% on the COCO2017 dataset, which is 1.5% and 1.6% higher than YOLOv5s, respectively. It has lower Params by 0.5M and lower FLOPs by 2.0G compared to YOLOv5s. It outperforms YOLOv7-Tiny-SiLU by 0.2% and 1.7%. Compared to YOLOv8n, the proposed algorithm achieves a 1.6% higher mAP, demonstrating superiority over lightweight versions of the YOLO series. Compared to larger models, DCGC-YOLO may have lower mAP, but it performs well on the fire dataset.

V. CONCLUSION

This paper proposes a lightweight object detection algorithm based on the improved YOLOv5s, specifically designed for fire image or video detection. Unlike YOLOv5s, the proposed algorithm adopts the DCGCLayer instead of the original C3 module. The DCGCLayer utilizes a CSP cascaded large convolution kernel structure to increase the network's receptive field, significantly enhancing feature extraction capabilities while reducing the model's parameter and computational complexity. Moreover, the eSE attention mechanism is employed to focus the model on learning relevant and informative features effectively. Lastly, the anchor sizes of the network are adjusted using the IoU-based k-means algorithm to better suit fire detection. Experimental results demonstrate that the proposed algorithm outperforms existing SOTA algorithms in terms of detection accuracy, and it requires fewer computational resources, both on self-built and public datasets.

Although the proposed algorithm achieves good detection accuracy, there are still instances of missed detections and false alarms. The robustness of the network may decrease when detecting objects similar to flames and smoke. In future work, we will further optimize the network and incorporate more objects similar to smoke and flames to enhance the network's detection efficiency.

REFERENCES

[1] S. Hall and B. Evarts, "Fire loss in the United States during 2021," Nat. Fire Protection Assoc., Quincy, MA, USA, Tech. Rep., 2021. [Online]. Available: https://www.nfpa.org/education-and-research/research/nfpa-research/fire-statistica l-reports/us-fire-department-profile

[2] D. Q. R. Elizalde, R. J. P. Garcia, M. M. S. Mitra, and R. G. Maramba, "Wireless automated fire detection system on utility posts using ATmega328P," in *Proc. IEEE 10th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ. Manage. (HNICEM)*, Nov. 2018, pp. 1–5.

[3] B. Kadri, B. Bouyeddou, and D. Moussaoui, "Early fire detection system using wireless sensor networks," in *Proc. Int. Conf. Appl. Smart Syst. (ICASS)*, Nov. 2018, pp. 1–4.

[4] A. R. Hutauruk, J. Pardede, P. Aritonang, R. F. Saragih, and A. Sagala, "Implementation of wireless sensor network as fire detector using Arduino nano," in *Proc. Int. Conf. Comput. Sci. Inf. Technol. (ICoSNIKOM)*, Nov. 2019, pp. 1–4.

[5] G. D. Raj and B. Prabadevi, "Steel strip quality assurance with YOLOV7-CSF: A coordinate attention and SIoU fusion approach," *IEEE Access*, vol. 11, pp. 129493–129506, 2023.

[6] G. Deepti Raj and B. Prabadevi, "Multiclass classification and defect detection of surfaces using modified-YOLO," in *Proc. 12th Int. Conf. Adv. Comput. (ICoAC)*, Aug. 2023, pp. 1–6.

[7] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11966–11976.

[8] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.

[9] Y. Lee and J. Park, "CenterMask: Real-time anchor-free instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13903–13912.

[10] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 12993–13000.

[11] T. Wang, P. Li, S. Fang, P. Zhang, Y. Yang, H. Liu, and L. Liu, "A multifunctional sensing and heating fabric based on carbon nanotubes conductive film," *IEEE Sensors J.*, vol. 23, no. 16, pp. 17990–17999, Jul. 2023.

[12] X. Liu, Z.-W. Tang, X.-C. An, Y.-L. Huang, Z.-H. Liu, Z. Tao, and J.-Y. Pan, "Infrared trace gas sensing with a fast perovskite nanostructure laser photodetector," *IEEE Photon. Technol. Lett.*, vol. 35, no. 1, pp. 19–22, Jan. 1, 2023.

[13] Y. Wang, Y. Han, Z. Tang, and P. Wang, "A fast video fire detection of irregular burning feature in fire-flame using in indoor fire sensing robots," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2022.

[14] F. Z. Rachman, G. Hendrantoro, and Wirawan, "Optimization of a fire detection system based on radial sector scanning using an UV sensor," *IEEE Sensors Lett.*, vol. 7, no. 7, pp. 1–4, Jul. 2023, Art. no. 5501904, doi: 10.1109/LSENS.2022.3225436.

[15] A. Gaur, A. Singh, A. Kumar, K. S. Kulkarni, S. Lala, K. Kapoor, V. Srivastava, A. Kumar, and S. C. Mukhopadhyay, "Fire sensing technologies: A review," *IEEE Sensors J.*, vol. 19, no. 9, pp. 3191–3202, May 2019.

[16] A. K. Bhandari, I. V. Kumar, and K. Srinivas, "Cuttlefish algorithm-based multilevel 3-D Otsu function for color image segmentation," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 5, pp. 1871–1880, May 2020.

[17] X. Chen, Q. An, K. Yu, and Y. Ban, "A novel fire identification algorithm based on improved color segmentation and enhanced feature data," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–15, 2021.

[18] M. Mueller, P. Karasev, I. Kolesov, and A. Tannenbaum, "Optical flow estimation for flame detection in videos," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2786–2797, Jul. 2013.

[19] T. Celik, "Fast and efficient method for fire detection using image processing," *ETRI J.*, vol. 32, no. 6, pp. 881–890, Dec. 2010.

[20] H.-C. Chang, Y.-L. Hsu, C.-Y. Hsiao, and Y.-F. Chen, "Design and implementation of an intelligent autonomous surveillance system for indoor environments," *IEEE Sensors J.*, vol. 21, no. 15, pp. 17335–17349, Aug. 2021.

[21] Z. S. A. Hakeem, H. I. Shahadi, and H. H. Abass, "An automatic system for detection of fires in outdoor areas," in *Proc. Int. Conf. Electr., Comput. Energy Technol. (ICECET)*, Jul. 2022, pp. 1–6.

[22] D. Pritam and J. H. Dewan, "Detection of fire using image processing techniques with LUV color space," in *Proc. 2nd Int. Conf. Converg. Technol. (I2CT)*, Apr. 2017, pp. 1158–1162.

[23] T. Qiu, Y. Yan, and G. Lu, "An autoadaptive edge-detection algorithm for flame and fire image processing," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 5, pp. 1486–1493, May 2012.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[25] W. Liu, D. Anguelov, D. Erhan, and C. Szegedy, "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf.* Cham, Switzerland: Springer, Oct. 2016, pp. 21–37.

[26] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[27] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[28] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

[29] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[30] 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.

[31] Ultralytics. (2023). *Ultralytics/YOLOv5: V7.0—YOLOv5 Sota Real-time Instance Segmentation*. [Online]. Available: https://github.com/ultralytics/yolov5.com

[32] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.

[33] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10778–10787.

[34] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, p. 217, Feb. 2021.

[35] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 7, pp. 1419–1434, Jul. 2019.

[36] L. Zhao, L. Zhi, C. Zhao, and W. Zheng, "Fire-YOLO: A small target object detection method for fire inspection," *Sustainability*, vol. 14, no. 9, p. 4930, Apr. 2022.

[37] S. S. Jandhyala, R. R. Jalleda, and D. M. Ravuri, "Forest fire classification and detection in aerial images using inception-V3 and SSD models," in *Proc. Int. Conf. Intell. Data Commun. Technol. Internet Things (IDCIoT)*, Jan. 2023, pp. 320–325.

[38] G. Lin, Y. Zhang, Q. Zhang, Y. Jia, G. Xu, and J. Wang, "Smoke detection in video sequences based on dynamic texture using volume local binary patterns," *KSII Trans. Int. Inf. Syst.*, vol. 11, no. 11, pp. 5522–5536, 2017.

[39] T. Toulouse, L. Rossi, A. Campana, T. Celik, and M. A. Akhloufi, "Computer vision for wildfire research: An evolving image dataset for processing and analysis," *Fire Saf. J.*, vol. 92, pp. 188–194, Sep. 2017.

[40] D. Y. T. Chino, L. P. S. Avalhais, J. F. Rodrigues, and A. J. M. Traina, "BoWFire: Detection of fire in still images by integrating pixel color and texture analysis," in *Proc. 28th SIBGRAPI Conf. Graph., Patterns Images*, Aug. 2015, pp. 95–102.

[41] J. Sharma and M. Goodwin. (2023). *Fire Detection Image Dataset*. [Online]. Available: https://github.com/cair/Fire-Detection-Image-Dataset

[42] M. Olafenwa. (2023). *Firenet*. [Online]. Available: https://github.com/OlafenwaMoses/FireNET

[43] A. Shamsoshoara, F. Afghah, A. Razi, L. Zheng, P. Fulé, and E. Blasch, "The flame dataset: Aerial imagery pile burn detection using drones (UAVs)," IEEE DataPort, New York, NY, USA, Rep., 2020. [Online]. Available: https://ieee-dataport.org/open-access/flame-dataset-aerial-imagery-pile-burn-detection-using-drones-uavs, doi: 10.21227/qad6-r683.

[44] A. Khan and B. Hassan. (2023). *Dataset for Forest Fire Detection*. [Online]. Available: https://data.mendeley.com/datasets/gjmr63rz2r/1

[45] D. Shah. (2023). *Fire Detection*. [Online]. Available: https://github.com/jackfrost1411/fire-detection

[46] G. Pereira, A. Fusioka, B. Nassu, and R. Minneto, "A large-scale dataset for active fire detection/segmentation (Landsat-8)," IEEE Dataport, Ningbo Univ., Zhejiang, China, Tech. Rep., 2020. [Online]. Available: https://ieee-dataport.org/documents/landsat-8-datasets-large-areas, doi: 10.21227/5m95-0e81.

[47] B. C. Ko, S. J. Ham, and J. Y. Nam, "Modeling and formalization of fuzzy finite automata for detection of irregular fire flames," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 12, pp. 1903–1912, Dec. 2011.

[48] P. Foggia, A. Saggese, and M. Vento, "Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 9, pp. 1545–1556, Sep. 2015.

[49] N. Bhowmik and T. P. Breckon, "Experimental exploration of compact convolutional neural network architectures for non-temporal real-time fire detection," in *Proc. 18th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2019, pp. 653–658.

[50] A. J. Dunnings and T. P. Breckon, "Experimentally defined convolutional neural network architecture variants for non-temporal real-time fire detection," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 1558–1562.

[51] R. Pande. (2023). *Fire Detection From CCTV*. [Online]. Available: https://www.kaggle.com/datasets/ritupande/fire-detection-from-cctv

[52] C. Kyrkou and T. Theocharides, "EmergencyNet: Efficient aerial image classification for drone-based emergency monitoring using atrous convolutional feature fusion," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 1687–1699, 2020.

[53] C. Kyrkou and T. Theocharides, "Deep-learning-based aerial image classification for emergency response applications using unmanned aerial vehicles," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 517–525.

[54] D. Labs. (2023). *Fire and Smoke Dataset*. [Online]. Available: https://www.kaggle.com/datasets/dataclusterlabs/fire-and-smoke-dataset

[55] P. F. Fule, A. W. Watts, F. A. Afghah, B. H. Hopkins, L. O. O'Neill, A. R. Razi, and J. C. Coen, "Flame 2: Fire detection and modeling: Aerial multi-spectral image dataset," School Comput., Clemson Univ., Clemson, SC, USA, Tech. Rep., 2022, doi: 10.21227/swyw-6j78.

[56] M. Wang, L. Jiang, P. Yue, D. Yu, and T. Tuo, "FASDD: An open-access 100,000-level flame and smoke detection dataset for deep learning in fire detection," *Syst. Sci. Data*, Jan. 2024, doi: 10.57760/sciencedb.j00104.00103.

[57] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[58] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9626–9635.

[59] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.

[60] K. Chen et al., "MMDetection: Open MMLab detection toolbox and benchmark," 2019, *arXiv:1906.07155*.

[61] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475.

[62] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

YUN HE received the B.S. degree in mechanical design and automation from Northeast Electric Power University, Jilin, China, in 2017. He is currently pursuing the M.S. degree in electronic information with Jishou University, Jishou, China. His research interests include computer vision and object detection.

JUNJIE HU received the B.S. degree in water resources and electric power from North China University, Henan, China. He is currently pursuing the M.S. degree in electronic information with Jishou University, Jishou, China. His research interests include computer vision, object detection, and deep learning. He has also been awarded the Campus Academic Scholarship.

MING ZENG received the B.S. degree in electrical engineering and automation from Huaihua University, Huaihua, China. He is currently pursuing the M.S. degree in electronic information with Jishou University, Jishou, China. His research interests include computer vision, object detection, and deep learning.

YINGJING QIAN received the B.S. degree in communications engineering from Nanchang University, Nanchang, China, in 2004, and the M.S. degree in circuits and systems from Hunan University, Changsha, China, in 2011. She is currently an Associate Professor with the School of Physics, Electronics and Intelligent Manufacturing, Huaihua University, and also a Postgraduate Supervisor with Jishou University. Her research interests include wireless communications and machine learning.

RENMIN ZHANG (Member, IEEE) received the B.S. degree in electric engineering from Guilin University of Electronic Technology, Guilin, China, in 2003, the M.S. degree in electronic and communication engineering from Peking University, Beijing, China, in 2011, and the Ph.D. degree in information and communication engineering from Southeast University, Nanjing, China, in 2019. He is currently a Professor with the School of Communication and Electronic Engineering, Jishou University. His research interests include signal and information processing.