

---

01 Jan 2022

## BCMNet: Cross-Layer Extraction Structure and Multiscale Downsampling Network with Bidirectional Transpose FPN for Fast Detection of Wildfire Smoke

Jiayong Li

Guoxiong Zhou

Aibin Chen

Chao Lu

*et. al.* For a complete list of authors, see [https://scholarsmine.mst.edu/civarc\\_enveng\\_facwork/2377](https://scholarsmine.mst.edu/civarc_enveng_facwork/2377)

Follow this and additional works at: [https://scholarsmine.mst.edu/civarc\\_enveng\\_facwork](https://scholarsmine.mst.edu/civarc_enveng_facwork)



Part of the [Architecture Commons](#), and the [Civil and Environmental Engineering Commons](#)

---

### Recommended Citation

J. Li et al., "BCMNet: Cross-Layer Extraction Structure and Multiscale Downsampling Network with Bidirectional Transpose FPN for Fast Detection of Wildfire Smoke," *IEEE Systems Journal*, Institute of Electrical and Electronics Engineers, Jan 2022.

The definitive version is available at <https://doi.org/10.1109/JSYST.2022.3193951>

This Article - Journal is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Civil, Architectural and Environmental Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact [scholarsmine@mst.edu](mailto:scholarsmine@mst.edu).

# BCMNet: Cross-Layer Extraction Structure and Multiscale Downsampling Network With Bidirectional Transpose FPN for Fast Detection of Wildfire Smoke

Jiayong Li , Guoxiong Zhou , Aibin Chen, Chao Lu, and LiuJun Li

**Abstract**—At present, the wildfire smoke detection algorithm based on YOLOv3 has problems, such as low accuracy and slow detection speed. In this article, we propose a cross-layer extraction structure and multiscale downsampling network with bidirectional transpose FPN (BCMNet) for fast detection of wildfire smoke. First, a cross-layer extraction module, which combines linear feature multiplexing and receptive field amplification, is designed. It can improve the speed and accuracy of wildfire smoke detection. Second, a multiscale downsampling module with different convolution kernels and maximum pooling operation is designed to preserve the details of the image while downsampling. Then, a bidirectional transposed FPN based on transposed convolution upsampling is designed. It can bidirectionally fuse visual features of shallow layer and semantic features of deep layer on the corresponding scale. The feature information flow between smoke feature maps of different resolution is emphasized. Finally, a wildfire smoke detection system of the Internet of Things based on BCMNet is built by combining the hardware and detection model. The experimental results show that the proposed method achieves 85.50% mAP<sup>50</sup> and 79.98% mAP<sup>75</sup> at 40 FPS on NVIDIA Geforce RTX 2080 Ti, which is superior to the common smoke detection methods.

**Index Terms**—Bidirectional transpose FPN (BTFPN), cross-layer extraction module (CLEM), cross-layer extraction structure and multiscale downsampling network with bidirectional transpose FPN (BCMNet), Internet of Things (IoT), multiscale downsampling module (MDSM), wildfire smoke detection.

## I. INTRODUCTION

**F**OREST is the main part of the terrestrial ecosystem, and plays an important role in regulating climate, conserving

Manuscript received 9 November 2021; revised 4 March 2022 and 9 May 2022; accepted 17 July 2022. This work was supported by the Scientific Research Project of Education Department of Hunan Province under Grant 21A0179, in part by Changsha Municipal Natural Science Foundation under Grant kq2014160, in part by the Natural Science Foundation of Hunan Province under Grant 2021JJ41087, and in part by Hunan Key Laboratory of Intelligent Logistics Technology under Grant 2019TP1015. (Corresponding author: Guoxiong Zhou.)

Jiayong Li, Guoxiong Zhou, Aibin Chen, and Chao Lu are with the Institute of Artificial Intelligence Applications, School of Computer Information and Engineering, Central South Forestry University of Technology, Changsha 410018, China (e-mail: jy892889422@gmail.com; t20060599@csuft.edu.cn; hotaibin@163.com; 1244414754@qq.com).

LiuJun Li is with the Department of Civil, Architectural and Environmental Engineering, University of Missouri, Rolla, MO 65409 USA (e-mail: lljwc@umsystem.edu).

Digital Object Identifier 10.1109/JSYST.2022.3193951

water, preventing wind and fixing sand, improving soil, and so on. Wildfire is a sudden, destructive, and uncontrollable natural disaster. Once a wildfire occurs, it will cause heavy losses of natural resources and human property. In the early stage of a fire, white smoke will be produced, and the diffusion of smoke is irregular and easily affected by environment, climate, and other factors, resulting in different concentrations, shapes, and sizes. Thus, it is difficult to identify wildfire smoke or the speed of smoke detection is slow. But, if we can detect this remarkable visual feature at an early stage, we can forewarn the fire in its cradle and reduce the loss to the minimum.

The early-day fire smoke detection systems mainly relied on traditional point sensors. When the space becomes larger, they cannot effectively detect smoke signals. In addition, because the sensor is susceptible to dust, airflow, and human factors, its detection efficiency is also greatly affected. With the development of computer vision and image processing technology, more and more researchers and institutions have begun to study smoke detection. Typical characteristics of smoke include color, texture, the direction of movement, etc. Chen et al. [1] mainly studied the color characteristics and diffusion characteristics of smoke and analyzed the rough distribution rules of smoke color in RGB three channels. They studied the dynamic characteristics of smoke diffusion movement, shape change, and growth rate to detect smoke. Krstinić et al. [2] used the HSI model to describe the color characteristics of smoke and enhance the separation of smoke and nonsmoke, which is better than RGB, YCbCr, CIE Lab, and HIS color models to a certain extent. However, many objects in the natural environment are similar to the color of smoke, so this method can only be used in the ideal environment. These methods are based on single characteristic. They have limited generalization ability and are greatly disturbed by the environment. Wang et al. [3] proposed a method for video smoke detection using visual features, such as shape, color, and dynamic texture of the image, which achieved good results. Zen et al. [4] used Gaussian mixture model to segment moving objects and then inputs the detection frame coordinates, area, and fog speed as feature parameters into the classifier to determine whether it is smoke.

With the rapid development of the Internet of Things (IoT) and deep learning technology, we can get more accurate and effective real-time features without the limitation of conventional pattern recognition methods. The IoT can connect all kinds of information sensing devices, such as infrared thermal imagers and high-definition video, unmanned aerial systems,

wireless sensor networks, radio frequency identification devices, personal digital assistants, and the Internet. They form a huge network [5], [6]. The IoT has become an important means to obtain accurate and quantitative information and has achieved great success in image processing. Therefore, the deep learning algorithm has been applied to smoke detection. Conventional smoke detection methods are based on shallow visual features, so they have poor generalization ability, cannot cope with the natural environment disturbance and the irregularity of smoke diffusion, and cannot be transplanted. Compared with the shallow feature, the deep semantic feature is more stable and representative. Yin et al. [7] proposed a new deep normalization and convolutional neural network (CNN) model with 14 layers to realize automatic feature extraction and classification. To speed up the training process and improve the performance of smoke detection, the network used a combination of normalization and convolution to replace the conventional convolutional layer. Cao et al. [8] proposed a novel attention-enhanced bidirectional long-term short-term memory network (LSTM) for video-based wildfire smoke recognition. It can not only capture the temporal and spatial characteristics of smoke in video frames but also use the attention mechanism to strengthen the temporal and spatial characteristics. Qiang et al. [9] analyzed the characteristics of smoke in each stage in detail and proposed a new method of wildfire smoke detection that combines robust principal component analysis in the time domain (TRPCA) and a dual-stream combination of VGG and BLSTM (TSVB) model. The dynamic and static characteristics of smoke are extracted from the time flow and spatial flow, and, finally, the two features are merged to realize the detection of wildfire smoke. This method achieves the accuracy of 90.6% on the self-built dataset, but the detection speed is slow. Shi et al. [10] used the “labeling” tool to label a dataset of fire and smoke and used YOLOv3 for detection. The results showed that YOLOv3 can better balance detection speed and accuracy. However, this structure will introduce excessive parameters while improving the feature extraction capability, increasing the detection time and calculation cost.

This article proposes a cross-layer extraction structure and multiscale downsampling network with bidirectional transpose FPN (BCMNet) for fast detection of wildfire smoke based on YOLOv3. In our implementation, the main contributions of this article are as follows.

- 1) We proposed a cross-layer extraction module (CLEM), which combines linear feature multiplexing and receptive field amplification. On the main road, a linear feature multiplexing module was designed to reduce the parameters of network training and the computing load of the processor. On the vice-road, a zigzag hybrid dilated convolution was designed to expand the receptive field. Different dilation rates were selected to improve the receptive field and ensure the integrity of smoke feature information. Finally, the main road and the vice-road are fused crossing layers, and it can improve the detection speed and accuracy of smoke object.
- 2) We proposed a multiscale downsampling module (MDSM). It preserves the details of the image while downsampling by using different-size convolution kernels and performing the maximum pooling operation in the three branches. It solves the problem of information loss caused by the traditional downsampling method.

Meanwhile, squeeze-and-excitation module (SE attention) was used to emphasize the object area of the smoke feature image and reduce the interference caused by the fuzzy and other redundant information.

- 3) We proposed a bidirectional transposed FPN (BTFPN) based on transposed convolution upsampling. It can bidirectionally fuse visual features of the shallow layer and semantic features after the deep transposed convolution layer on the corresponding scale. Then, solve the problem that FPN can only integrate the deep semantic features into the shallow visual feature information in one direction, which leads to the insufficient utilization of feature information, and improve the accuracy of smoke object detection.

The structure of this article is as follows. Section II briefly introduces the related work of smoke recognition. Section III introduces BCMNet for fast detection of wildfire smoke. Section IV conducts test verification and result analysis. Finally, Section V concludes this article.

## II. RELATED WORK

In the field of object detection, conventional methods are divided into the following three steps:

- 1) information area selection;
- 2) feature extraction;
- 3) classification and positioning.

Cheng et al. [11] proposed a regional contrast based salient object detection algorithm, which calculates both the global contrast difference and the spatially weighted coherence. This algorithm is simple and efficient and can generate multiscale, high-resolution and high-quality maps. Lowe et al. [12] proposed a scale-invariant feature transform algorithm in 2004. This algorithm seeks the extreme point in space and extracts the location of local features, which has good robustness. Zhou et al. [13] proposed to use low-rank to represent objects with regional continuity, and this model solved this problem by combining the spatial distribution information of moving objects. However, due to the greedy nature of decolor algorithm, the background around moving objects would be misdetected as the foreground. At present, the object detection algorithms based on deep learning are mainly divided into two categories: one-stage object detection algorithms and two-stage object detection algorithms. Two-stage object detection algorithms are: faster R-CNN, mask R-CNN, and FPN. Girshick et al. [14] proposed the R-CNN algorithm in 2014, which improved the average accuracy (mAP) by more than 30% compared to the previous best results of VOC 2012. The algorithm applied high-capacity CNN to the candidate region to locate and classify objects. In 2015, Faster R-CNN [15] algorithm was proposed. It is a fast convolutional network method based on candidate regions. Based on the previous work of R-CNN, Faster R-CNN adopted several innovations to improve the accuracy of object detection and the training speed of the network was also improved. Lin et al. [16] proposed the FPN algorithm in 2016. This algorithm combines high-level semantic information and low-level semantic information to detect objects of different scales in different feature layers after fusion, thus forming a pyramid network structure, optimizing the accuracy of object detection and solving the problem of multiscale object detection. He et al. [17] proposed the mask

R-CNN algorithm in 2018, which generates high-quality segmentation masks for each object. It extends the faster R-CNN algorithm by adding a parallel branch for predicting the object mask to an existing branch for object recognition. A two-stage algorithm often has higher accuracy, but there is a problem that the model structure is complex and the detection speed is slow. A one-stage algorithm can improve detection speed and higher real-time due to simplifying screening and optimization of the predictive box. The current object detection algorithm is developed for mobile and lightweight development, so the lightweight one-stage object detection algorithm is a hot trend of current research. Liu et al. [18] proposed a method of detecting objects in images with a single neural network in 2016 named SSD, that is, a one-stage object detection algorithm. The SSD algorithm is simple compared to the network that needs to generate candidate regions because it eliminates the calculation of feature resampling and generating a large number of candidate regions, is easy to train, and can be directly integrated into other systems. This algorithm achieved 74.3% of the mAP on the VOC2007 test dataset. The research on an anchor is only an important direction of object detection, which is mainly divided into anchor-free and anchor-based. The earliest anchor-based theory from the faster R-CNN refers to the designation of a set of anchor boxes before training and then selection according to the size of the object by clustering. Lin et al. [19] proposed the RetinaNet in 2017. Its design focal loss can effectively solve the problem of imbalance between positive and negative samples caused by the anchor-based algorithm. In fact, most objects are irregular, so it is quite complicated to design an anchor manually in advance, and a large number of anchors will lead to the problem of sample imbalance. Therefore, the algorithm based on anchor-free appears. Law et al. [20] proposed the CornerNet in 2018. It no longer uses the preset anchor, but to determine the object through top-left and bottom-right points, at the same time, the author also proposed a corner pooling. CornerNet reaches 42.2% of AP on the MSCOCO dataset, but the detection speed is low. The advantage of an anchor-free algorithm is that its greater and more flexible solution space and the accuracy of the object box is higher theoretically. However, these algorithms can only generate one object box for each key point. When the object coincidence degree in the graph is high, there will be missed detection. YOLO series was originally published in 2016. YOLOv1 [21] is a typical anchor-free algorithm and the earliest one-stage object detection algorithm, with a very high detection speed. YOLOv2 [22] uses anchor boxes and the K-means algorithm for clustering. YOLOv3 algorithm [23] was proposed by Redmon and Farhadi in 2018. It used the DarkNet-53 network structure. Integrated the idea of the FPN algorithm, it predicted three different scales of boxes. Similarly, integrated multilayer feature information solved the problem of poor results of a one-stage object detection algorithm in small object detection. YOLOv4 [24] adds SPP, squeeze-and-excitation (SE), soft NMS, DIoU NMS, etc., based on YOLOv3, and still uses YOLOv3's head section. But YOLOv4's Backbone is even more huge, which does not meet lightweight and rapid development trends to a certain extent. To detect early fire smoke quickly and reduce the property loss and environmental harm caused by fire, this article proposes BCMNet based on the task framework of YOLOv3. As shown in Fig. 1, the network mainly includes feature extraction, feature fusion, and boundary box prediction. We have improved the feature extraction and feature fusion to realize fast and accurate smoke detection.

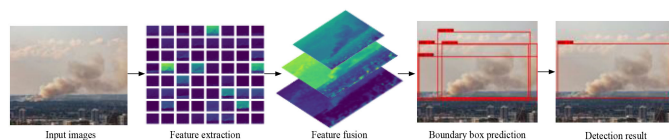


Fig. 1. Schematic diagram of BCMNet.

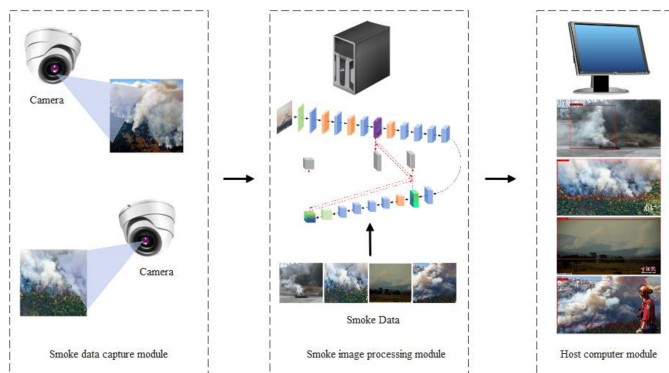


Fig. 2. Schematic diagram of the wildfire smoke detection system of the IoT based on BCMNet.

### III. MATERIALS AND METHODS

#### A. Wildfire Smoke Detection System of IoT Based on BCMNet

This article combines BCMNet and hardware equipment to build an IoT system of wildfire smoke detection based on BCMNet. The system mainly includes the smoke data acquisition module, smoke image processing module, and upper computer module. Smoke images were captured by using a high-definition pan-head camera called Hikvision DS-2DYH277I-DU, which has 2 million pixels and a resolution of  $1920 \times 1080$ . It is capable of shooting in both horizontal  $0^\circ \sim 360^\circ$  and vertical  $0^\circ \sim 45^\circ$  directions. Shooting parameters are set and images are acquired continuously in the surveillance area of the camera. Then, the collected image data is transmitted to the server through the network for image processing and object detection. The detection results can be viewed in real-time on the upper computer. The schematic diagram of the system is shown in Fig. 2.

#### B. Data Acquisition

The original dataset of this article consists of the following three parts.

- 1) *Web crawler*. We search from Baidu pictures, Google pictures, Bing pictures, and other image search engines according to keywords (smoke, forest fire, and wildfire), and use the crawler program to automatically download the pictures in the search results.
- 2) *Field shooting*. We artificially ignite wildfire burning in Zhuzhou Forest Farm, then use HikVision DS-2DYH277I-DU to shoot related smoke images.
- 3) *Obtained through public dataset* [25]. Then, we manually filter the original dataset to remove irrelevant images, flame images, duplicate images, similar images, blurred images, and low-resolution images.

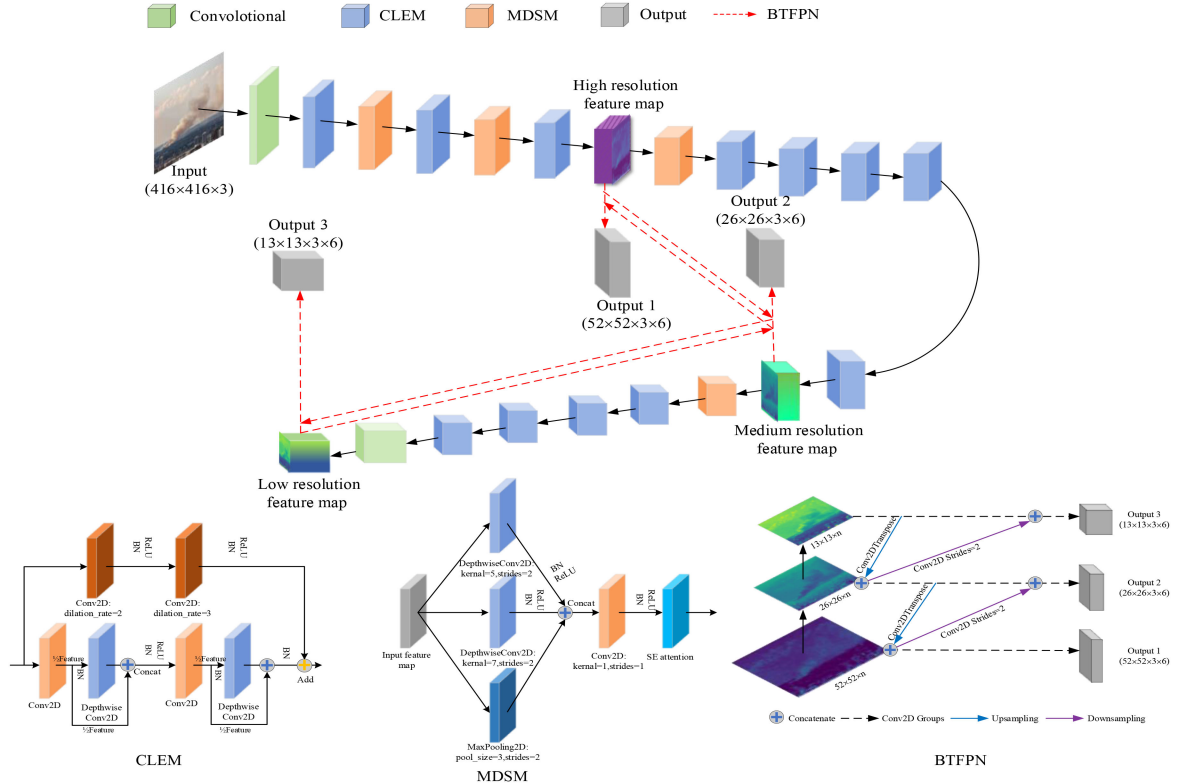


Fig. 3. BCMNet structure diagram.

Finally, we collected a total of 1206 smoke images, forming the wildfire smoke dataset of this article. The types and quantities of smoke in this dataset are shown in Table I. The smoke in column a is clear and of moderate size, which can be easily identified by conventional detection methods. Column b shows that the smoke object is so large that it is difficult to accurately locate it on the map. Column c shows that the shape of the smoke is small and the distance is far, which is difficult for human vision to detect. Column d shows that smoke concentration is small and the texture, color, edge, and other features are not obvious. Column e shows that the smoke is of high concentration and appears in yellow or even black, which is quite different from normal smoke. To sum up, due to the problem that the diffusion of smoke is irregular, the concentration, shape and size are changeable in the natural environment and it is difficult for common smoke detection methods to accurately detect the smoke in columns b–e. Therefore, it is necessary to propose a wildfire smoke detection algorithm to deal with the images of smoke with irregular diffusion.

### C. BCMNet for Fast Detection of Wildfire Smoke

Section III-B shows that wildfire smoke in the natural environment is easily affected by the environment, climate, and other factors due to its irregular diffusion, resulting in different concentrations, shapes, and sizes of the smoke. Therefore, wildfire smoke detection is difficult. YOLOv3 algorithm has low accuracy and slow detection speed. Therefore, we propose a BCMNet for fast detection of wildfire smoke, which can better balance the accuracy and detection speed of smoke detection. The structure of BCMNet is shown in Fig. 3. The part connected

by the black arrow is the feature extraction part, which is mainly composed of 12 CLEMs and 4 MSDMs. The part connected by the red arrow is the feature fusion part, and the structure is BTFPN.

1) *Feature Extraction*: In the feature extraction stage, we designed a new backbone, which is mainly composed of CLEM and MDSM. CLEM is an efficient smoke feature extraction structure, which can effectively increase the size of the receptive field. MDSM can retain the detailed information of the image during the downsampling process. Meanwhile, SE Attention was adopted to enhance the saliency of the object area of the smoke feature map. The parameter setting of BCMNet's backbone is shown in Fig. 4.

a) *CLEM*: In the conventional neural network [26]–[30], the large channel dimension can increase the expression of the feature information, but the larger the channel dimension, the heavier the learning burden of the neural network [31], [32]. So, the channel dimension is the key factor affecting the performance of the network. In the overall design of CLEM, we first mapped the input of the low-dimensional channel to the high-dimensional channel inside CLEM to enrich the expression of smoke characteristics. Then, in the output part of CLEM, we compressed the high-dimensional channel to reduce the dimension and reduce the number of parameters to ensure the overall faster calculation speed of BCMNet.

Visualize the features after passing through the first residual structure of YOLOv3 as shown in Fig. 5 and we can find several features that are very similar. When the neural network further analyzes these similar features, it will obtain a lot of redundant information, which is actually unnecessary. Therefore, only some of the features will be further convoluted to extract deeper

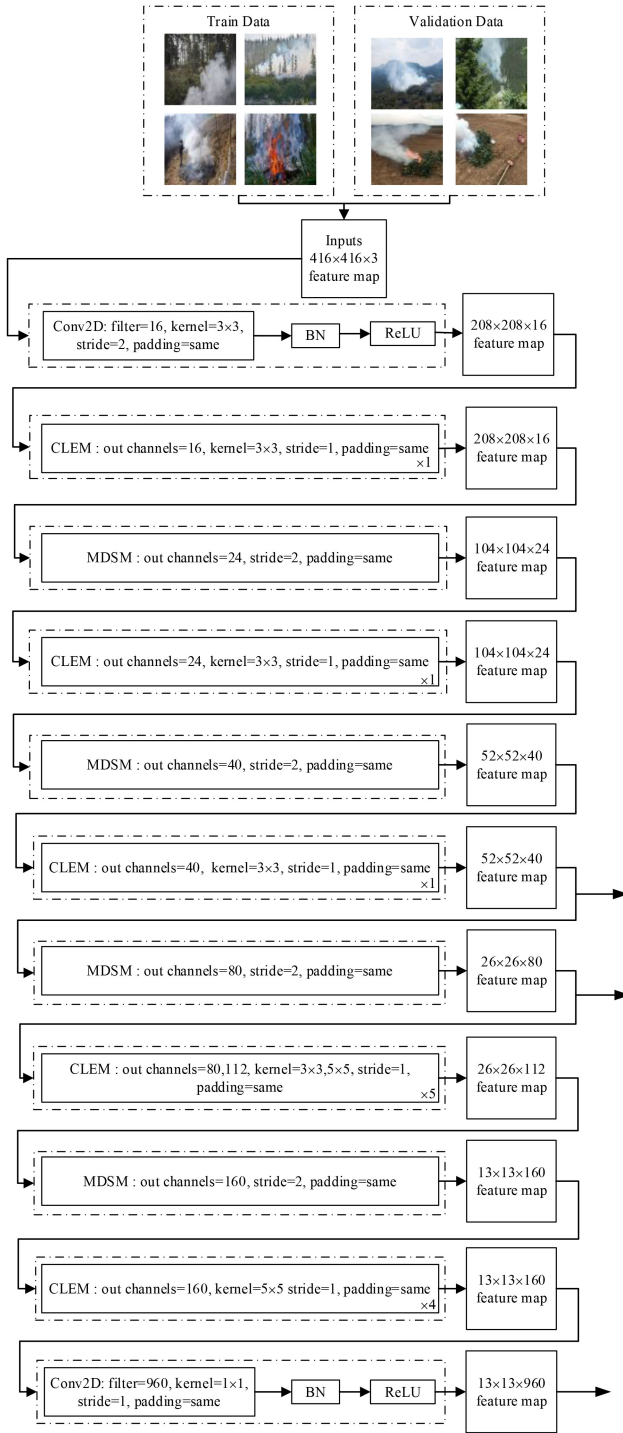


Fig. 4. Backbone's structure parameter settings.

information, whereas the other similar features will be directly fused with deeper features, which can not only ensure that the neural network has a more comprehensive understanding of the input data but also reduce the computational burden. So, we designed a linear feature multiplexing structure on the main road of CLEM, and the structure of CLEM is shown in Fig. 3. We only perform convolution operations on half of the high-dimensional channels in CLEM to get the benchmark feature map  $Y$ , then perform linear deep convolution operations on  $Y$  to generate

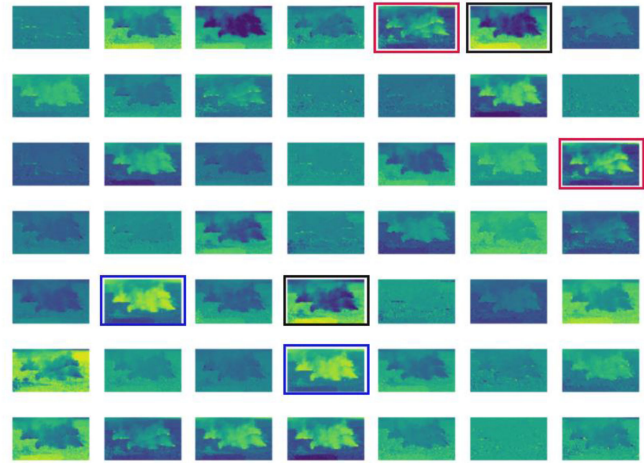


Fig. 5. Features behind the first residual structure of YOLOv3.

feature maps  $Y'$  of the same channel dimension, and then multiplex the  $Y$  feature and linearly fuse with  $Y'$  to form the feature graph  $W$  with a high-dimensional channel. The formula is as follows:

$$Y = X * f + b \quad (1)$$

where  $X \in \mathbb{R}^{c \times h \times w}$  is the input feature map with channel number  $c$ , width  $w$ , and height  $h$ ,  $*$  is the convolution operation,  $b$  is the bias term,  $f \in \mathbb{R}^{c \times k \times k \times n}$  is the convolution kernel of this layer, and  $Y \in \mathbb{R}^{n \times h \times w}$  is the benchmark feature map with the number of channels  $n$ . Then,  $Y$  is calculated to get the new feature graph given as

$$Y' = \varphi(Y) + b \quad (2)$$

where  $\varphi$  is the linear convolution operation and  $Y' \in \mathbb{R}^{n \times h \times w}$  is the input feature map with channel number  $n$ . Finally,  $Y$  and  $Y'$  are merged to obtain a high-dimensional feature map  $W$ , and the formula is as follows:

$$W = \phi[Y, Y'] \quad (3)$$

where  $\phi$  is the fusion of channel dimensions and  $W \in \mathbb{R}^{2 \times n \times h \times w}$  is the high-dimensional feature map after fusion.

Compared with the conventional neural network using convolution to increase the channel dimension, the linear feature multiplexing structure can effectively improve detection speed. Besides, the feature map generated by linear operation has higher information relevance. It is beneficial to the supplementary expression of smoke characteristic information. On the vice-road, we designed a receptive field amplification module using a zigzag hybrid cavity convolution combination. This kind of zigzag expansion convolution uses different expansion rates, which can effectively maintain the continuity of the smoke characteristic information during continuous convolution, improve the receptive field, and ensure the integrity of the smoke characteristic information. Then, a cross-layer fusion of the main road and branch road will further increase the receptive field of CLEM. CLEM can well retain the spatial characteristics of the smoke, and will not lose the characteristic information of the smoke. We can intuitively deduce the size of the receptive field amplification through the formula, and the calculation method

for the receptive field of the  $n$ th layer is

$$l_n = l_{n-1} + \left( (k_n - 1) * \prod_{i=1}^{n-1} s_i \right) \quad (4)$$

where  $l_{n-1}$  is the size of the receptive field of the  $(n-1)$ th layer,  $k_n$  is the size of the convolution kernel of the  $n$ th layer, and  $s_i$  is the step size of the  $i$ th layer. It can be calculated by formula (4) that the receptive field between the two layers increases by  $((k_n - 1) * \prod_{i=1}^{n-1} s_i)$ .

We use dilated convolution, and the increase in expansion rate can be intuitively reflected by transforming the convolution kernel, and then the convolution kernel in the expanded convolution layer can be transformed into

$$k_{\text{trans}} = (r - 1) * (k - 1) + k \quad (5)$$

where  $r$  is the dilate rate, and  $k$  is the actual convolution kernel of the current layer.

*b) MDSM:* Downsampling is one of the important operations in neural networks. It can proportionally reduce the size of the feature map, increase the receptive field, simplify the computational complexity of the network, and prevent overfitting. Pooling is the most common type of downsampling. It can filter the features in the receptive domain and extract the most significant features in the area. However, pooling will cause the network model to lose translation invariance. Meanwhile, for smoke objects with blurred edges, multiple downsampling will cause some information such as smoke posture and spatial position in the image to be lost, resulting in inaccurate positioning. On the other hand, pooling is a static operation that cannot be learned, and it is difficult to deal with smoke detection with complex interference factors and changeable posture. Therefore, we proposed MDSM, which can retain the detailed information of the image while downsampling, and its structure is shown in Fig. 3. We split the input into three branches for different operations. Since the deep convolution with less computation is used, a larger convolution kernel can be used to increase the receptive field. Therefore, in the first branch and the second branch, the convolution kernel of  $5 \times 5$  and  $7 \times 7$  with strides of 2 are, respectively, used to carry out the convolution operation to reflect the smoke characteristics more comprehensively. In the third branch, a maximum pooling operation with strides of 2 is carried out to extract more significant smoke characteristics. Then the outputs of the three branches are fused to further enrich the smoke feature information. Then, a  $1 \times 1$  convolution is used to compress the channel and reduce the dimensionality. Meanwhile, the multiscale smoke feature information extracted from the three channels is integrated. Finally, to ensure that the module can pay more attention to the characteristics of smoke, SE attention [33] was adopted to enhance the saliency of the object area of the smoke feature map, and reduce the interference caused by redundant information such as blurring of the smoke image. SE attention is a channel-based attention mechanism, which can be divided into two stages: Squeeze and Excitation.

The squeezing process is as follows:

Global average pooling for the feature with an input of  $w \times h \times C$ , and get a feature map of  $1 \times 1 \times C$  with a global

receptive field. For the  $i$ th channel, there is

$$z_i = \frac{1}{w \times h} \sum_{p=1}^W \sum_{q=1}^H u_i(p, q) \quad (6)$$

where  $w \times h$  represents the resolution of the original feature map;  $u_i(p, q)$  represents the element whose coordinates of the  $i$ th channel layer is  $(p, q)$ , and the total number of channels is  $C$ ; and  $z_i$  is the feature mapping amount of the channel. The one-dimensional vector  $z \in \mathbb{R}^C$  of  $1 \times 1 \times C$  is obtained through this compression process. The Excitation process is as follow:

$$s = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)). \quad (7)$$

First, reduce the number of channels to  $C/r$  of the original amount through a pointwise convolution layer with weight  $W_1$ , where  $r$  is the reduction ratio. Then, enter the pointwise convolution layer with weight  $W_2$  to restore the channel dimension after ReLU ( $\delta$ ) activation, and, finally, use the Sigmoid function ( $\sigma$ ) to generate normalization channel weight  $s \in \mathbb{R}^C$ . The scale is  $1 \times 1 \times C$ . Multiply the normalized channel weight and the corresponding channel of the original feature map to obtain the channel attention feature map.

*2) Feature Fusion:* FPN [34] is the most popular feature fusion method among current object detection algorithms. It can represent objects of different scales and improve the accuracy of detection. The original YOLOv3 designed a one-way fusion method, which can simply merge the smoke features from top to bottom and integrate the smoke features of different scales. However, this one-way fusion method does not fully utilize features, so we designed a BTFPN, as shown in Fig. 3. First of all, in the upsampling part, a transposed convolution with kernel size =  $3 \times 3$  and strides = 2 was used instead of the original interpolation method in YOLOv3. It has the advantages of low information redundancy and a strong ability of mapping features of smoke, which can ensure the consistency of original smoke feature information. In the downsampling part, a strided convolution with strides = 2 and kernel size =  $3 \times 3$  was used. The feature map is reduced by half in the spatial dimension. Finally, the shallow visual features and deep semantic features were bidirectionally fused in the channel dimension, which emphasized the feature information flow between smoke feature maps of different resolutions. The detection and positioning accuracy of smoke objects were improved.

*3) Loss Function:* BCMNet's loss function is the same as YOLOv3. The loss function of BCMNet is divided into three parts: the object location offset loss  $L_{\text{loc}}(l, g)$ , the target confidence loss  $L_{\text{conf}}(o, c)$ , and the object classification loss  $L_{\text{cla}}(O, C)$

$$L(O, o, C, c, l, g) = L_{\text{conf}}(o, c) + L_{\text{cla}}(O, C) + L_{\text{loc}}(l, g). \quad (8)$$

The formula to calculate  $L_{\text{conf}}(o, c)$  is as follows:

$$L_{\text{conf}}(o, c) = - \sum (o_i \ln(\hat{c}_i) + (1 - o_i) \ln(1 - \hat{c}_i)) \quad (9)$$

$$\hat{c}_i = \text{Sigmoid}(c_i) \quad (10)$$

where  $o_i \in \{0, 1\}$  and  $\hat{c}_i$  represents the Sigmoid probability that there is an object in the prediction object boundary box  $i$ .

The formula to calculate  $L_{\text{cla}}(O, C)$  is as follows:

TABLE I  
WILDFIRE SMOKE DATASET STATISTICS






Group	a	b	c	d	e
Image					
Number	408	297	146	163	192

TABLE II  
HARDWARE AND SOFTWARE PARAMETERS

Hardware environment	CPU	Intel Core i9-10980XE
	RAM	64G
	GPU	NVIDIA GeForce RTX 2080 Ti
	Video memory	16G
Software environment	OS	Windows 10
		CUDA Toolkit 10.0; CUDNN V7.5.0; Python 3.6; Tensorflow-GPU 1.14

TABLE III  
PARAMETER SETTING OF BCMNET

Input image	Batch size	Epoch	Optimizer	Base learning rate
416×416	16	1000	Adam	0.001

TABLE IV  
INFLUENCE OF DIFFERENT INPUT SIZE ON BCMNET

Input size	$mAP^{50}(\%)$	$mAP^{75}(\%)$	FPS
BCMNet (416×416)	85.50	79.98	40
BCMNet (512×512)	86.81	81.22	36
BCMNet (608×608)	88.42	83.79	33

TABLE V  
COMPARISON OF MODEL EVALUATION

Method	$mAP^{50}(\%)$	$mAP^{75}(\%)$	AR	FPS	Params	FLOPs
YOLOv3	74.39	66.36	40.50	33	62M	308M
BCMNet	85.50	79.98	46.16	40	23M	109M

$$L_{cla}(O, C) = - \sum_{i \in Pos} \sum_{i \in cla} (O_{ij} \ln(\hat{C}_{ij}) + (1 - O_{ij}) \ln(1 - \hat{C}_{ij})) \quad (11)$$

$$\hat{C}_{ij} = \text{Sigmoid}(C_{ij}) \quad (12)$$

where  $o_{ij} \in \{0, 1\}$  represents whether there is actually the  $j$ th category of the object in the prediction object boundary box  $i$ .

The formula to calculate  $L_{loc}(l, g)$  is as follows:

$$L_{loc}(l, g) = \sum_{i \in Pos} \sum_{m \in (x, y, w, h)} (\hat{l}_i^m - \hat{g}_i^m)^2 \quad (13)$$

where  $\hat{l}$  represents the coordinate offset of the prediction rectangular box.  $\hat{g}$  represents the coordinate offset between the matching ground true box and the default box.

## IV. EXPERIMENTAL ANALYSIS

### A. Experimental Environment

To verify the performance of BCMNET proposed in this article, all experiments in this article are run in the same hardware and software environment. The specific environmental parameters are shown in Table II.

In this article, we first shuffle the dataset, and then use tenfold cross-validation to train the model. For all input images, we re-size them to  $416 \times 416$ . To accelerate the convergence speed and improve the stability of the model, an Adam optimizer was used in this article and a cosine annealing algorithm was used to set the learning rate. For the first 400 epochs, the maximum learning rate was set at  $1e-3$  and the minimum was  $1e-6$ . In the subsequent training, the maximum learning rate was set at  $1e-4$  and the minimum was  $1e-6$ . We use K-means to cluster wildfire smoke dataset labels, get the anchor of nine groups of different sizes: (62,64), (107,96), (182,198), (273,166), (431,372), (481,247), (667,404), (833,733), and (1817,1790). Batch normalization layer uses Keras default hyperparameters and the momentum is 0.99. Other training parameter settings are shown in Table III.

### B. Evaluation Index

In this article, the performance of the model is evaluated by precision ( $P$ ), recall ( $R$ ), mAP, AR, FPS, parameter size, and FLOPs.

We divide the test results into the following categories: precision ( $P$ ) is the proportion of correct classification. Recall ( $R$ ) is the ratio of the amount of relevant information detected to the total amount. The formula is defined as

$$P = \frac{TP}{TP + FP} \quad (14)$$

$$R = \frac{TP}{TP + FN} \quad (15)$$

where  $T_P$  is true positive,  $F_P$  is false positive,  $T_N$  is true negative, and  $F_N$  is false negative.

Mean average accuracy (mAP) is a quantitative indicator to evaluate the detection effect of multicategory objects. The formula to calculate mAP is as follows:

$$mAP = \frac{\sum_{i=1}^k AP_i}{k} \quad (16)$$

$$AP = \sum_{i=1}^n P(i) \Delta r(i) \quad (17)$$

where  $k$  is the class number,  $P(i)$  is the Precision at the threshold  $i$ , and  $\Delta r(i)$  is the change in recall between  $i$  and  $i+1$ .



TABLE VI  
INDIVIDUAL PERFORMANCE STATISTICS FOR EACH TYPE OF SMOKE

Group	a	b	c	d	e
YOLOv3/mAP <sup>50</sup>	100%	77.52%	26.27%	64.62%	100%
BCMNet /mAP <sup>50</sup>	100%	86.13%	60%	83.43%	100%

TABLE VII  
VISUAL COMPARISON OF TEST RESULTS

Experimental method	Detection result				
YOLOv3					
CLEM					
CLEM+BTFPN					
BCMNet					
	a	b	c	d	e

Average recall (AR) is an indicator of the missed detection of the detector. The formula to calculate AR is as follows:

$$AR = \frac{\text{Recall}}{n} \quad (18)$$

where  $n$  is the number of detected object frames.

Frames per second (FPS) is an important indicator to measure the detection speed. The formula to calculate FPS is as follows:

$$FPS = \frac{1}{t} \quad (19)$$

where  $t$  is the time required to process each picture frame.

Parameter size and FLOPs are indicators used to measure the complexity of the model.

### C. Performance of the BCMNet Method

To demonstrate the effectiveness of our approach, we conducted a series of tracking experiments on the same dataset. To evaluate our network model comprehensively, we figured out the experimental results of a series of indicators such as mAP, FPS, and AR. We discuss the performance of BCMNet in image resolution commonly used in object detection, as shown in Table IV. The input size of the image affects the accuracy and speed of wildfire smoke detection. The higher the image resolution, the accuracy of the detection will be improved, but the detection speed will also be reduced. To compare more rigorous

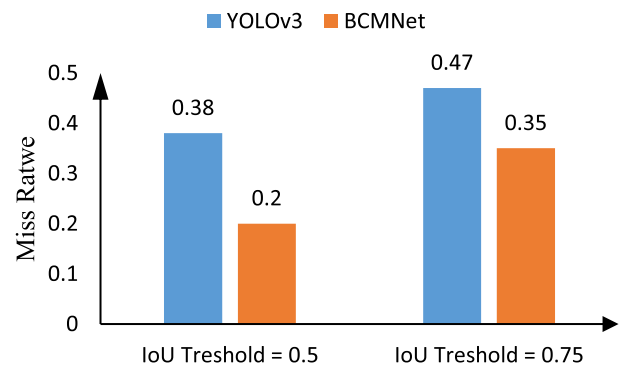


Fig. 6. Comparison of miss rate under different IOU.

experiments with other methods, we set the input size of all models to  $416 \times 416$  in addition to Section IV-F.

First, this article compared the detection effects of two network models under different IOU thresholds, as shown in Table V. Regarding accuracy, mAP<sup>50</sup> and mAP<sup>75</sup> of our model were, respectively, 11.11% and 13.62% higher than that of YOLOv3. The AR has increased by 5.66%. Those indicated the effectiveness of BCMNET in feature extraction and feature fusion. Regarding the detection speed, our model reached 40 FPS, which not only met the requirement of real-time detection but was far faster than the detection speed of YOLOv3. Moreover, the

TABLE VIII  
ABLATION EXPERIMENT RESULTS

Number	Method	mAP <sup>50</sup> (%)	mAP <sup>75</sup> (%)	AR(%)	FPS	Param	FLOPs
1	CLEM+MDSM+BTFPN (BCMNet)	85.50	79.98	46.16	40	23M	109M
2	CLEM+MDSM+ YOLO_FPN	84.67	71.41	45.56	43	22M	106M
3	CLEM + BTFPN + DarknetConv2D_BN_Leaky	83.91	65.13	45.62	43	23M	109M
4	MDSM+BTFPN+ Res_block	81.32	68.26	45.91	35	60M	286M
5	CLEM + DarknetConv2D_BN_Leaky+YOLO_FPN	81.46	79.09	45.17	47	22M	107M
6	MDSM + Res_block+ YOLO_FPN	77.40	72.33	42.76	37	58M	282M
7	BTFPN + Res_block +DarknetConv2D_BN_Leaky	79.46	78.43	42.78	32	64M	312M
8	Res_block +DarknetConv2D_BN_Leaky+ YOLO_FPN (YOLOv3)	74.39	66.36	40.50	33	62M	308M

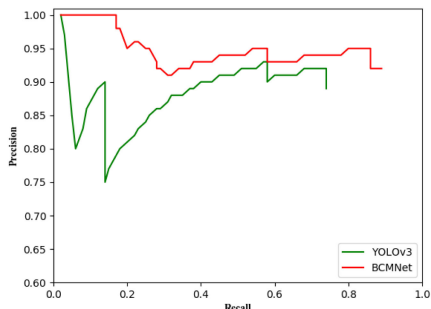


Fig. 7. PR curve at IoU threshold = 0.5.

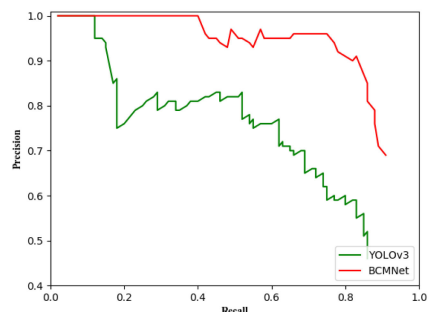


Fig. 8. PR curve at IoU threshold = 0.75.

parameters and FLOPs of BCMNet were only 37% and 35% of those of YOLOv3. This is because CLEM adopts the linear feature multiplexing structure and reasonable channel design strategy, which has fewer parameters.

Then, this article compared the miss rate of BCMNet with YOLOv3. The histogram of the smoke detection loss rate of the two models is shown in Fig. 6. Under the two thresholds, our loss rates are both lower. The PR curves of the two network models on the wildfire smoke dataset are shown in Figs. 7 and 8. It can be seen that BCMNet performs better.

We separated the different types of smoke in the test set for statistics, then calculated mAP for each case, and finally, compared the results with YOLOv3, as shown in Table VI. To analyze BCMNet more intuitively, we visualize the detection results of YOLOv3 and the network model designed by us. As shown in Table VII, the smoke frame, category and confidence level are displayed on the detection result graph.

In group a, the smoke objects are clear and of moderate size. Both models perform very well. Compared with YOLOv3's

confidence of 96%, our methods all achieved 100%. It can be seen that the positioning of BCMNet is more accurate than that of YOLOv3.

The smoke object in the image of group b is huge and it is difficult to locate it. Compared to YOLOv3, BCMNet is not only more accurate in positioning, but also has a 25% increase in confidence. This is because CLEM has a larger receptive field and can better retain the spatial characteristics of large object smoke.

In group c, the smoke object is small and far away. Our method can effectively monitor small-scale smoke objects. On the one hand, this is because of the strong feature extraction capability of CLEM. On the other hand, we adopted MDSM, which will not lose smoke information in the continuous process of downsampling and can effectively retain the salient features of smoke objects of different scales. Therefore, for small object smoke, BCMNet's mAP is 33.73% higher than YOLOv3. In group d, the concentration of smoke objects was low, and the shallow visual characteristics were not obvious. YOLOv3 cannot detect it, whereas the confidence of our method reached 100%. At the same time, BCMNet increases mAP by 19.81%. This is because BTFPN combines the shallow visual features and deep semantic features of smoke to enhance the significance of smoke features.

In group e, the smoke concentration was high, and the color was yellow or even black, which was different from the conventional gray-white smoke. Although all the methods can detect the smoke object, our method is more accurate and has higher confidence. This not only reflects the excellent feature extraction ability of BCMNet but also shows its powerful generalization ability.

It can be seen from the five groups of experimental results that BCMNET's performance is very excellent, especially for small objectives and objects with no obvious characteristics.

#### D. Ablation Experiment

To verify the effectiveness of the method proposed in this article, we conducted verification experiments on each innovation point in the framework of YOLOv3 and conducted ablation experiments of BCMNet.

The settings and results of the ablation experiment are shown in Table VIII. Based on BCMNet, we used control variables to remove CLEM, MDSM, and BTFPN one by one, and replace them with Res\_block (residual block), DarknetConv2D\_BN\_Leaky (followed by Conv2D, batch normalization, and leaky ReLU operations), and YOLO\_FPN (FPN structure in YOLOv3) with corresponding functions in YOLOv3. Among them, the first group is BCMNet, and the eighth group is YOLOv3.

TABLE IX  
COMPARISON OF DETECTION EFFECTS OF BCMNET MODEL AND OTHER NETWORKS

Method	Input size	backbone	mAP <sup>50</sup> (%)	FPS
Faster R-CNN	224 × 224	VGG16	64.21	10
SSD300	300 × 300	VGG16	66.74	32
SSD513	513 × 513	ResNet101	69.07	14
YOLOv3	416 × 416	ResNet50	73.33	33
YOLOv3	416 × 416	Darknet53	74.39	33
RetinaNet	512 × 512	ResNet101	82.45	15
ComerNet-Lite [35]	255 × 255	SqueezeNet	82.61	35
YOLOv4	416 × 416	CSPDarknet53	84.97	34
YOLOF [36]	416 × 416	ResNet101	84.71	36
YOLOR [37]	416 × 416	CSP	85.35	38
YOLOv5-s [38]	416 × 416	Focus + CSPDarknet53	70.62	68
YOLOv5-x [38]	416 × 416	Focus + CSPDarknet53	87.66	20
YOLOX-s [39]	416 × 416	Modified CSP v5	71.15	51
YOLOX-x [39]	416 × 416	Modified CSP v5	88.02	16
BCMNet	416 × 416	-	85.50	40

The comparison between group 7 and group 8 shows that the BTFPN structure can improve the index mAP of smoke detection while it increased the amount of model calculation to a small extent and reduced 1 FPS. This is because of the increment of parameters while upsampling using transposed convolution instead of interpolation. The comparison between group 6 and group 8 shows that MDSM is effective in improving both mAP and FPS. This is because MDSM can retain smoke details during the downsampling process, and on the other hand, the use of depthwise convolution reduced parameter calculation. A comparison of group 5 and group 8 shows that CLEM increased the indexes mAP<sup>50</sup> and mAP<sup>75</sup> by 7.07% and 12.73%, and FPS increased by 14, which proves that the channel design strategy of CLEM and multiplexing of linear characteristics helps improve real-time performance while getting a lightweight model. It can keep the integrity of smoke information while expanding the receptive field. The comparison results of the eight-group experiments can fully prove that CLEM, MDSM, and BTFPN have better extraction capabilities for smoke characteristics, and BCMNet has higher detection speed and accuracy than YOLOv3.

#### E. Comparison Experiment of Different Detection Models

To verify the performance of the BCMNet model, we compared it with some classic or advanced object detection methods under the same dataset. The result is shown in Table IX.

In the same input size, BCMNet is better than other algorithms in the YOLO series. Compared to YOLOv3, YOLOv4, YOLOF, and YOLOR, BCMNet is better in detection accuracy and detection speed. YOLOv5-s and YOLOX-s have very fast detection speeds, but the mAP is 14.88% and 14.35% lower than BCMNet, respectively. Although the mAP of YOLOv5-x and YOLOX-x is higher than the BCMNet, their detection speed is very low, which is 20FPS and 24FPS lower than BCMNet. Compared with other classical target detection algorithms (faster R-CNN, SSD series, RetinaNet, and Cornernet-Lite), the detection speed and accuracy of BCMNet are also better.

Therefore, through the experimental comparison, it can be concluded that BCMNet can not only meet the requirements of

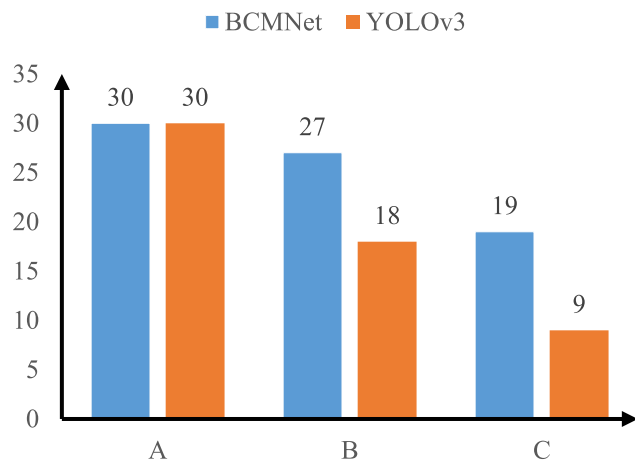


Fig. 9. Experimental comparison. Class A refers to the situation where the smoke density and size are moderate. Class B refers to the situation where the smoke density is small and the characteristics are not obvious. Class C refers to the situation where the smoke object is small and difficult to find.

real-time testing but also have a very high accuracy rate for wildfire smoke detection. BCMNet is an excellent wildfire detection algorithm that can balance detection speed and accuracy.

#### F. Testing of Real Applications

We conducted a simulation experiment of wildfire combustion at Zhuzhou Forest Farm. To ensure the integrity of image information, we used the original resolution (1920 × 1080) taken by the Hikvision DS-2DYH277I-DU camera as the input size. In the case of high resolution, the detection speed of BCMNet can reach 28FPS to meet real-time detection requirements. In addition, we simulated 30 times of smoke in three states and compared the number of smoke recognition using YOLOv3 and BCMNet. As can be seen from Fig. 9, BCMNet has a higher recognition rate for wildfire smoke and provides more effective ideas for wildfire prevention.

We show some smoke images that are difficult to detect, as shown in Fig. 10. Fig. 10(a) shows the situation where the smoke

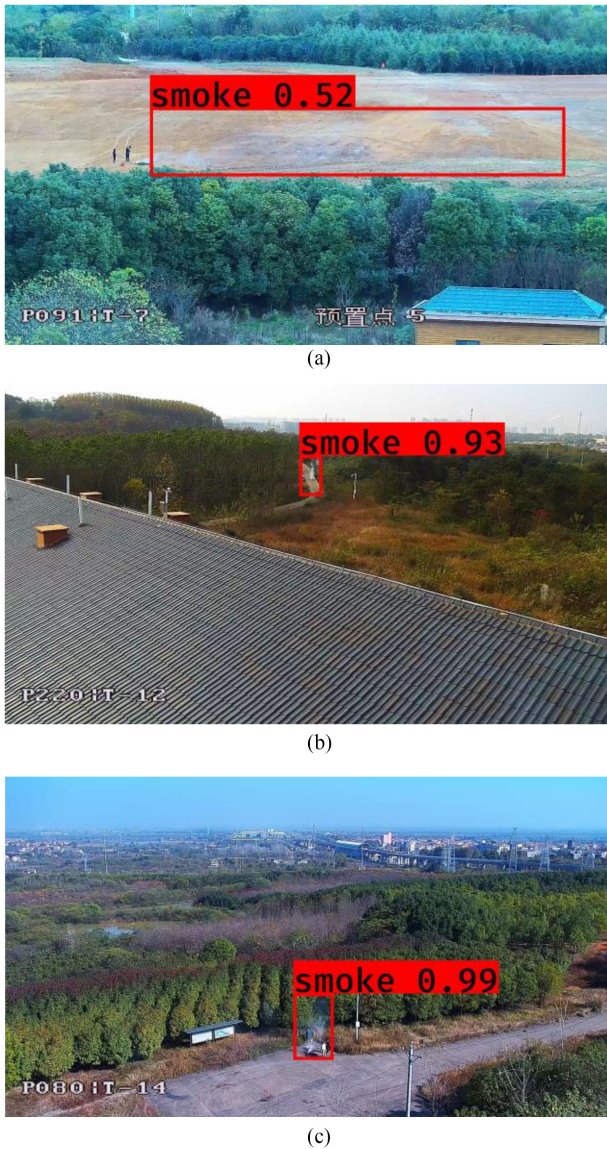


Fig. 10. Real applications. (a) Low concentration smoke image. (b) Remote smoke image. (c) Small object smoke image.

area is large but the concentration is small, and the method proposed in this article can effectively detect it. Fig. 10(b) shows the case where the smoke object is small, and our method can also detect small objects effectively. In Fig. 10(c), the smoke object is small and its concentration is low, and BCMNet can detect it well. It fully demonstrates the excellence of BCMNet for smoke detection tasks.

## V. CONCLUSION

To improve the effectiveness and the speed of detection of smoke, a BCMNet for fast detection of wildfire smoke is proposed in this article. Compared with the YOLOv3, the improvements of the method proposed in this article are as follows.

- 1) BCMNet has reached 40 FPS on NVIDIA Geforce RTX 2080. This is because the linear feature multiplexing structure and reasonable channel strategy are widely used in CLEM.

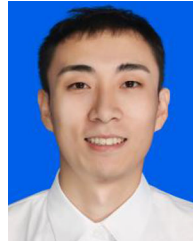
- 2) On the self-built smoke dataset, the proposed BCMNet achieves 85.50% mAP<sup>50</sup>, 79.98% mAP<sup>75</sup>, and 46.16% AR. Visualization of test results and comparison of experiments show that BCMNet can effectively improve the accuracy of smoke detection. This is because CLEM has a larger receptive field. Meanwhile, MDSM can ensure the integrity of smoke feature information during the downsampling operation. BTFPN can bidirectionally fuse visual features of the shallow layer and semantic features of the deep layer on the corresponding scale. The feature information flow between smoke feature maps of different resolutions is emphasized.
- 3) This article designs a wildfire smoke detection system based on BCMNet, and it is used in practice. It can detect smoke and predict the occurrence of wildfire in time and accurately, which is of great significance to protect ecological resources and reduce losses.

Although BCMNet has a good performance in smoke detection, it still has a high false detection rate for smoke-like objects, such as lens flares and water droplets, on the lens. In the next step, we will in-depth study the difference between the characteristics of smoke and these interference objects, and further, improve the accuracy while ensuring a faster detection speed. At the same time, our research will not be limited to the detection of visible light images, and we will further explore some other types of smoke images for wildfire detection, such as infrared images and hyperspectral images.

## REFERENCES

- [1] C. Chen, W. D. Zhong, and D. Wu, "Color multiplexing based unipolar OFDM for indoor RGB LED visible light communication," *Procedia Eng.*, vol. 140, pp. 159–165, 2016.
- [2] D. Krstinić, D. Stipančević, and T. Jakovčević, "Histogram-based smoke segmentation in forest fire detection system," *Inf. Technol. Control*, vol. 38, no. 3, pp. 237–244, 2009.
- [3] S. Wang, Y. He, H. Yang, K. Wang, and J. Wang, "Video smoke detection using shape, color, and dynamic features," *J. Intell. Fuzzy Syst.*, vol. 33, no. 1, pp. 305–313, 2017.
- [4] R. I. Zen, M. R. Widyanto, G. Kiswanto, G. Dharsono, and Y. S. Nugroho, "Dangerous smoke classification using mathematical model of meaning," *Procedia Eng.*, vol. 62, pp. 963–971, 2013.
- [5] J. E. Siegel, S. Kumar, and S. E. Sarma, "The future internet of things: Secure, efficient, and model-based," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2386–2398, Apr. 2017.
- [6] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8182–8201, May 2019.
- [7] Z. Yin, B. Wan, F. Yuan, X. Xia, and J. Shi, "A deep normalization and convolutional neural network for image smoke detection," *IEEE Access*, vol. 5, pp. 18429–18438, 2017.
- [8] Y. Cao, F. Yang Y., Q. Tang, and X. Lu, "An attention enhanced bidirectional LSTM for early forest fire smoke recognition," *IEEE Access*, vol. 7, pp. 154732–154742, 2019.
- [9] X. Qiang, G. Zhou, A. Chen, X. Zhang, and W. Zhang, "Forest fire smoke detection under complex backgrounds using TRPCA and TSVB," *Int. J. Wildland Fire*, vol. 30, no. 5, pp. 329–350, 2021.
- [10] F. Shi, H. Qian, W. Chen, M. Huang, and Z. Wan, "A fire monitoring and alarm system based on YOLOv3 with OHEM," in *Proc. IEEE 39th Chin. Control Conf.*, 2020, pp. 7322–7327.
- [11] M. M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S. M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, Mar. 2014.
- [12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [13] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 597–610, Mar. 2013.

- [14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.
- [15] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.
- [16] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks detection for object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern*, 2017, pp. 2117–2125.
- [17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.
- [18] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
- [19] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, "Automatic ship detection based on RetinaNet using multi-resolution Gaofen-3 imagery," *Remote Sens.*, vol. 11, no. 5, 2019, Art. no. 531.
- [20] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 734–750.
- [21] 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.*, 2016, pp. 779–788.
- [22] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7263–7271.
- [23] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
- [24] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
- [25] Q. X. Zhang, G. H. Lin, Y. M. Zhang, G. Xu, and J. J. Wang, "Wildland forest fire smoke detection based on faster R-CNN using synthetic smoke images," *Procedia Eng.*, vol. 211, pp. 441–446, 2018.
- [26] G. Zhou, W. Zhang, A. Chen, M. He, and X. Ma, "Rapid detection of rice disease based on FCM-KM and faster R-CNN fusion," *IEEE Access*, vol. 7, pp. 143190–143206, 2019.
- [27] W. Zhang, J. Hu, G. Zhou, and M. He, "Detection of apple defects based on the FCM-NPGA and a multivariate image analysis," *IEEE Access*, vol. 8, pp. 38833–38845, 2020.
- [28] X. Chen, G. Zhou, A. Chen, J. Yi, W. Zhang, and Y. Hu, "Identification of tomato leaf diseases based on combination of ABCK-BWTR and B-ARNet," *Comput. Electron. Agriculture*, vol. 178, 2020, Art. no. 105730.
- [29] A. Tan, G. Zhou, and M. He, "Surface defect identification of citrus based on KF-2D-Renyi and ABC-SVM," *Multimedia Tools Appl.*, vol. 80, no. 6, pp. 9109–9136, 2021.
- [30] W. Zhang et al., "A method for classifying citrus surface defects based on machine vision," *J. Food Meas. Characterization*, vol. 15, pp. 2877–2888, 2021.
- [31] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1580–1589.
- [32] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.
- [33] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.
- [34] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.
- [35] H. Law, Y. Teng, O. Russakovsky, and J. Deng, "Cornernet-lite: Efficient keypoint based object detection," 2019, *arXiv:1904.08900*.
- [36] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, and J. Sun, "You only look one-level feature," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13039–13048.
- [37] 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*.
- [38] G. Jocher, 2021. [Online]. Available: <https://github.com/ultralytics/yolov5>
- [39] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding Yolo series in 2021," 2021, *arXiv:2107.08430*.



**Jiayong Li** received the B.Sc. degree in computer science and technology (major) from Shandong Agricultural University, Tai'an, China, in 2018. He is currently working toward the M.Sc. degree in computer application technology with Central South University of Forestry and Technology, Changsha, China.

His research interests include deep learning and graphics and image processing.



**Guoxiong Zhou** received the Ph.D. degree in control science and engineering from Central South University, Changsha, China, in 2010.

He is currently an Associate Professor with Central South University of Forestry and Technology, Changsha, China. His research interests include forest fire prevention and robotics.



**Aibin Chen** received the Ph.D. degree in computer application technology from Central South University, Changsha, China, in 2010.

He is currently a Professor with Central South University of Forestry and Technology, Changsha, China. His main research interests include artificial intelligence and forest information engineering.



**Chao Lu** received the B.Sc. degree in software engineering (major) from Anhui Normal University, Wuhu, China, in 2019. He is currently working toward the M.Sc. degree in software engineering with Central South University of Forestry and Technology, Changsha, China.

His main research interests include deep learning and graphics and image processing.



**Liujun Li** received the B.Sc. degree in automation from Hunan Agricultural University, Changsha, China, in 2002, and the M.Sc. degree in materials engineering and the Ph.D. degree in mechanical engineering from Central South University, Changsha, China, in 2005 and 2012, respectively.