













Research Article

Fire-PPYOLOE: An Efficient Forest Fire Detector for Real-Time Wild Forest Fire Monitoring

Pei Yu ¹, Wei Wei ², Jing Li ², Qiuyang Du ^{1,3}, Fang Wang ², Lili Zhang ², Huitao Li ², Kang Yang ², Xudong Yang ², Ning Zhang ², Yucheng Han ² and Huapeng Yu ⁴

¹China Fire and Rescue Institute, Beijing 102202, China

²Beijing Institute of Petrochemical Technology, Beijing 102617, China

³Key Laboratory of Forest and Grassland Fire Risk Prevention, Ministry of Emergency Management, Beijing 102202, China

⁴Institute of National Defense Science and Technology Innovation, Academy of Military Sciences, Beijing 100036, China

Correspondence should be addressed to Wei Wei; weiwei@bipt.edu.cn and Jing Li; bipt_lijing@bipt.edu.cn

Received 12 December 2022; Revised 18 September 2023; Accepted 13 December 2023; Published 18 January 2024

Academic Editor: Yunchao Tang

Copyright © 2024 Pei Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Forest fire has the characteristics of sudden and destructive, which threatens safety of people's life and property. Automatic detection and early warning of forest fire in the early stage is very important for protecting forest resources and reducing disaster losses. Unmanned forest fire monitoring is one popular way of forest fire automatic detection. However, the actual forest environment is complex and diverse, and the vision image is affected by various factors easily such as geographical location, seasons, cloudy weather, day and night, etc. In this paper, we propose a novel fire detection method called Fire-PPYOLOE. We design a new backbone and neck structure leveraging large kernel convolution to capture a large arrange area of reception field based on the existing fast and accurate object detection model PP-YOLOE. In addition, our model maintains the high-speed performance of the single-stage detection model and reduces model parameters by using CSPNet significantly. Extensive experiments are conducted to show the effectiveness of Fire-PPYOLOE from the views of detection accuracy and speed. The results show that our Fire-PPYOLOE is able to detect the smoke- and flame-like objects because it can learn features around the object to be detected. It can provide real-time forest fire prevention and early detection.

1. Introduction

In the past few decades, frequency and scale of global wildfire have increased dramatically [1]. We take China alone as an example. From January to November 2021, 546 forest fires and 17 grassland fires occurred in China. Global forest fires are characterized by prolonged fire duration, expanded fire scope, and serious release of harmful gases, which affect social order and threaten heritage security [2].

Forest fire is very harmful and difficult to dispose and rescue. Therefore, forest fire monitoring, as an effective means of forest fire prevention and spread control, has become a major global research topic. Traditional forest fire monitoring is mainly based on observation tower patrol aircraft or satellite remote sensing images. However, this reditional way of forest fire monitoring by the weather climate technology level and monitoring of operating costs

does not provide forest fire forecast information in real time. With the rapid development of science and technology, manned aircraft inspection and unmanned aerial vehicle (UAV) inspection monitoring has become a more effective means of forest fire monitoring. It has the advantages of high efficiency, low cost, and strong real-time performance [3].

The traditional smoke fire detection methods mainly focus on the feature extraction and classification of static pictures or dynamic videos. The typical features of smoke contain color, texture, motion orientation, etc. [4]. Wang [5] designed a forest fire monitoring system using principal component analysis dimensionality reduction method to analyze the specificity of each channel of the three-color spaces.

With the widespread use of deep learning in target recognition and image classification in recent years, more and more researchers have started to combine this method with forest fire forecast tasks. Convolutional neural network

(CNN) was first used in smoke and fire image classification [4, 6–8]. In general, the CNN or R-CNN outperforms other machine learning methods, such as support vector machine, stack autoencoder, and deep belief network, in terms of classification accuracy, receiver operating characteristic curve, recall rate, and F1-score [6]. CNN has good detection accuracy for small objects, but the target of the flame may be large due to the shooting distance. Therefore, the YOLO method, an one-stage target detection algorithm, is proposed to improve the global detection accuracy and reduce the error detection rate. Wu et al. [9] proposed to combine the CNN Deeplab V3 + model with classical image processing algorithms to finely segment the beams and calculate the number of beams. The whole cluster of banana fruit was identified based on deep learning. The edge detection algorithm was used to extract the centroid of fruit finger shape, and the clustering algorithm was used to determine the optimal number of fruit bundles on the visual detection plane. The accuracy of beam detection in debudding stage was 86%. During the harvest, beam detection is very challenging, with a detection accuracy of 76%. Chen et al. [10] proposed an optimal YOLO-v4 detection method for bayberry trees based on drone images. Speed up model extraction by using the Leaky_ReLU activation function and use DIoU NMS to retain the high accuracy prediction boxes. The optimal YOLO-v4 model had a detection accuracy of up to 97.78% and a recall rate of up to 98.16% on the dataset. Li et al. [11] proposed a remote sensing image detection (RSI-YOLO) method based on YOLOv5 object detection algorithm. Channel attention and spatial attention mechanisms are used to enhance the features of neural network fusion. The multiscale feature fusion structure based on PANet is improved to a weighted bidirectional feature pyramid structure. In addition, the loss function is modified to optimize the network model. Jiao et al. [12] has proposed a deep learning fire detection algorithm that aims to improve the accuracy and efficiency of fire detection using drones. Extensive studies on fire detection using large-scale YOLOv3 and tiny-YOLOv3 network have been shown to be capable of learning representative and have presented ideal detection accuracy, about 91%, and the frame rate can reach up to 30 frames per second (FPS). Zhao et al. [13] proposed an improved fire-yolo deep learning algorithm. By extending the feature extraction network in three dimensions, the feature propagation ability of small fire target identification is enhanced, the network performance is improved, and the model parameters are reduced. Furthermore, through the enhancement of the feature pyramid, the best performance prediction box is obtained. The average detection time of the real-time model is 0.04 s per frame.

To solve the problem of low accuracy of early fire image recognition based on single-stage target detection model, the following improvements are made in this paper:

- (1) The feature extraction capability of backbone is improved by using large-core convolution instead of ordinary convolution kernel to improve the accuracy of early fire image recognition.
- (2) The CSPNet network is introduced to reduce the model parameters so as to reduce the resources consumed by model reasoning.

- (3) The network structure is changed and the reasoning speed is greatly increased to solve the problem of slow reasoning speed caused by large kernel convolution.

The main parts of this paper are structured as follows. Section 2 introduces PP-YOLOE briefly and then elaborates the improvement of Fire-PPYOLOE. To test the performance of the model proposed in this paper, the results of the three models on labeled and unlabeled datasets are compared and analyzed in Section 3. In addition, Section 3 gives some experimental details, and the conclusion is provided in Section 4.

2. Materials and Methods

For practical applications, there are high requirements for forest fire detection models, such as fast detection speed, high recall, low computing cost, and deployment of multiple application devices. This paper develops an efficient real-time forest fire detection model based on state-of-the-art object detection model PP-YOLOE [14] and name it as Fire-PPYOLOE. In this section, we introduce PP-YOLOE briefly and then elaborate the improvements of Fire-PPYOLOE.

2.1. The Overview of PP-YOLOE. PP-YOLOE is an efficient single-stage anchor-free model [15] based on PPYOLOv2 [16]. It introduced several updates such as scalable backbone and neck structure, task alignment learning [17], and efficient task-aligned head. It has excellent recall and speed performance compared with PP-YOLOv2. For example, the PP-YOLOE-large (<https://github.com/PaddlePaddle/PaddleDetection/tree/release/2.4/configs/ppyoloe>) achieves 51.6 mAP on COCO test-dev 2017 dataset and 78.1 FPS on Tesla V100. The speed can be further increased by about 100% using TensorRT FP16.

To meet the high requirements of forest fire monitoring, we design a new backbone and neck structure based on large kernel convolutions. The proposed Fire-PPYOLOE can further improve the detection recall and decrease the computing cost without sacrificing the detection speed.

2.2. The Proposed Fire-PPYOLOE. It is necessary to preprocess the images captured by monitoring devices to facilitate deep neural network calculations [18]. First, we normalize the image and map the value of color channel from [0–255] to [0–1]. Then, we adjust the image size to a uniform scale (e.g., 640×640). Next, the preprocessed image will be passed directly to our fire detector Fire-PPYOLOE. There is no need to locate candidate object region based on predefined anchors because our model is anchor-free. This will improve the detection speed to some extent. As shown in Figure 1, the proposed Fire-PPYOLOE is able to detect multiple flames in one time. Once the flames are detected, the result will be transmitted to the terminal device such as UAV.

The image is sent to the target detection network for prediction in the following steps. The first step is to put the image into backbone for feature extraction at different scales. Then, the feature map is put into the head (detection layer) to predict the location and category of the target. We

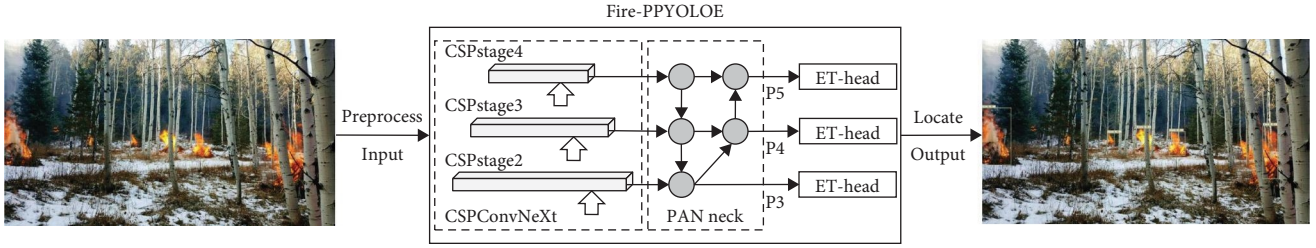


FIGURE 1: Flowchart of fire detection.

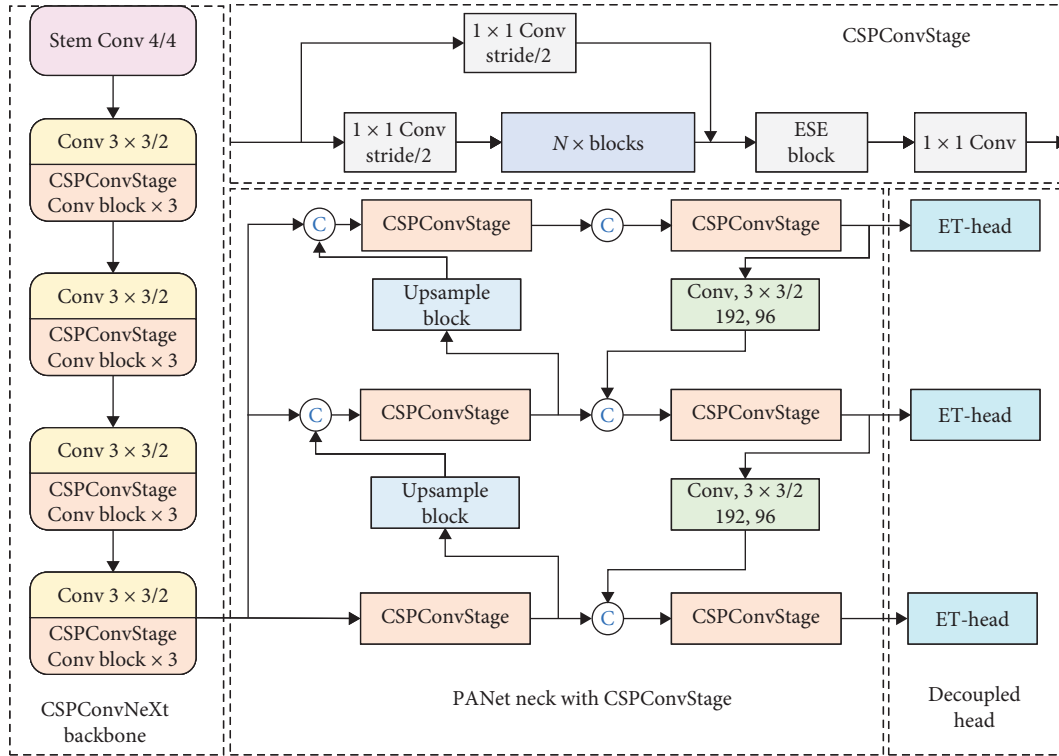


FIGURE 2: Model framework of Fire-PPYOLOE.

select multiscale feature maps for target prediction and add neck part to fuse different scale feature maps to improve the recognition accuracy of small objects. The feature map with higher scale has a strong feature extraction ability for image details. It can recognize small targets better.

To meet the practical usage, the fire detector should be able to be deployed on different endpoint equipments. Therefore, we train tiny/small/large models to cover different scenarios in practice. In this paper, we take the large version as an running example. The network structure of our Fire-PPYOLOE is shown in Figure 2. It consists of three parts, namely, the new designed backbone with ConvNeXt [19] and CSPNet [20] structure, the PANet [21] neck with CSPConvStage, and the efficient ET-head used in the original PP-YOLOE. We will introduce them in detail in the following subsections.

2.2.1. CSPConvNeXt Backbone. In the original PP-YOLOE, CSPResNet [14] is used as the backbone to extract multiple dimensional features. It leverages many 3×3 convolution

layers for feature extraction. The receptive field calculation formula is shown in Equation (1) [22]. We can see that the kernel size and the network depth are positively correlated with the size of receptive field

$$L_k = L_{k-1} + \left((f_k - 1) * \prod_{i=1}^{k-1} S_i \right), \quad (1)$$

where L represents the size of receptive field, f represents the kernel size, and S represents the stride. The receptive field size is equal to its kernel size.

Theoretically, CSPResNet can fully extract the features at every position. However, this is not the case. The effective receptive field (ERF) [22] is proposed to show the effective area of L_k , where Figure 3 (This picture is referred from Scaling Up Your Kernels to 31×31 : Revisiting Large Kernel Design in CNNs [23]) shows the effective areas of different networks with various kernel sizes. Equation (2) shows its computation formula. It has been proven that deep neural

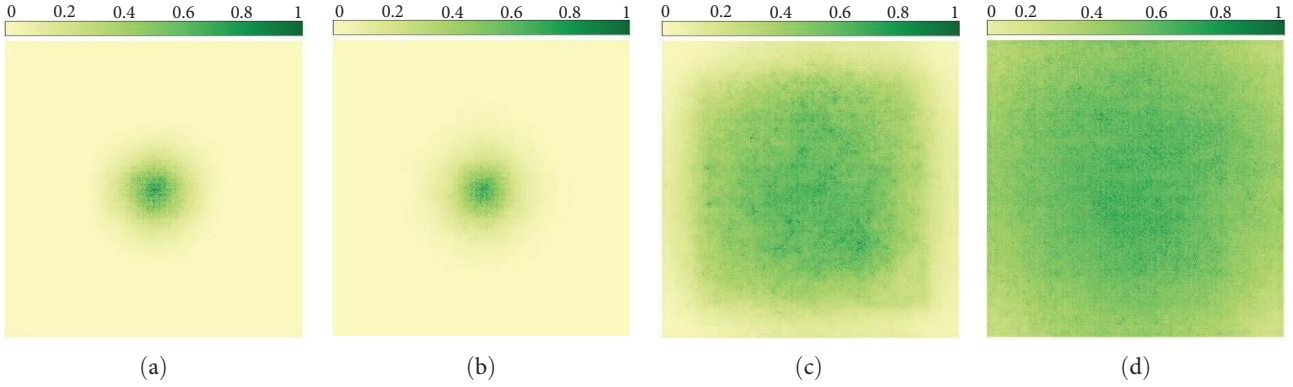


FIGURE 3: Effective receptive field of four models. (a) ResNet-101, (b) ResNet-152, (c) RepLKNet-13, and (d) RepLKNet-31.

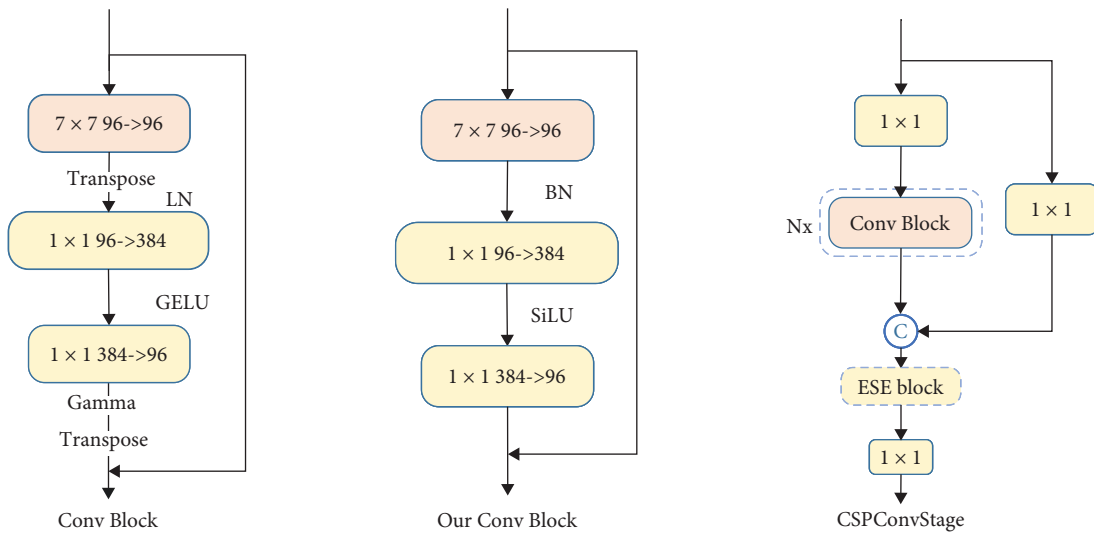


FIGURE 4: Backbone network details of CSPConvNeXt.

networks using small kernel convolutions pay more attention to the central part of the image and ignore the feature extraction for the edge part [23]

$$\text{ERF} = K\sqrt{L}. \quad (2)$$

From Equation (2), we can see that ERF is positively related to kernel size, and it is better to improve ERF by increasing the kernel size K compared with network depth L . Inspired by this, we propose to leverage large kernel convolutions to extract better ERF, so as to improve the detection recall. Although the speed of the large kernel convolution network is faster than that of the network formed by attention mechanisms, it still has a big disadvantage of detection speed compared with the traditional convolution network.

In this paper, we choose to use ConvNeXt [24] to replace CSPResNet and propose to introduce CSPNet to improve the recall without speed damage. As shown in Figure 2, the overall structure of the network starts with a stem with a kernel size of four and stride of four. Then, there are four structures, which consist of CSPConvStage and convolutions with a

kernel size of three and stride of two. For the large version, the width and depth of every CSPConvStage layer are [96,192,384,768] and [3,3,9,3], respectively. The feature map of the last three structures will be output. We make some improvements to backbone.

Figure 4 shows the details of the network structure improvement. We change layer normalization [25] in the original ConvNeXt block to batch normalization [26] and further remove gamma to improve speed. In addition, the activation function GELU [27] is changed to a powerful function SiLU [28]. Most importantly, we leverage CSPNet to optimize network structure. Compared with the CSPResNet, the network parameters are greatly reduced, and the recall is also improved. Some specific parameters will be shown in the experimental section.

2.3. PANet Neck with CSPConvStage. Neck is a network structure to fuse the extracted features from backbone. The original PP-YOLOE uses Path Aggregation Network (PANet) [21] as the neck. Our Fire-PPYOLOE updates PANet using the same CSPConvStage with its backbone. PANet with CSPConvStage has a large receptive field by changing the small kernel



FIGURE 5: Examples of fire images in labeled dataset.

convolution to large kernel convolution. Therefore, it can fully integrate features of different dimensions extracted by the backbone network, so as to improve the detection recall. As shown in the neck structure in Figure 2, it uses up sampling and down sampling modules to fuse the features of each layer and output the fused feature map to the head module.

2.4. ET-Head. The role of head is to predict the location and object class. The ET-head used in PP-YOLOE is proved to be very efficient, so we use ET-head directly in Fire-PPYOLOE. Varifocal loss (VFL) [29] and distribution focal loss (DFL) [30] are used to improve the recall and speed. Specifically, VFL uses the target score to weight the loss of positive samples, and this makes the contribution of positive samples with high intersection over union (IOU) to loss relatively large. It also makes the model pay more attention to the high-quality samples rather than the low-quality samples during the training process. This can effectively learn a joint representation of classification score and localization quality estimation, such that there is a high degree of consistency between training and inference. Therefore, VFL can make up the imbalance of positive and negative samples in forest fire detection. Equation (3) shows its computation formula:

$$\text{VFL}(p, q) = \begin{cases} -q(q \log(p) + (1 - q) \log(1 - p)) & q > 0 \\ -\alpha p^r \log(1 - p) & q = 0 \end{cases}, \quad (3)$$

where p is the predicted IOU-aware classification score and q is the target score.

DFL proposes to solve the problem of inflexible bounding boxes by using conventional distribution prediction bounding boxes

$$\text{DFL}(S_i, S_{i+1}) = -((y_{i+1} - y) \log(S_i) + (y - y_i) \log(S_{i+1})), \quad (4)$$

where y represents the regressed label and S represents the softmax function.

Based on the above computation, Fire-PPYOLOE is supervised by the following loss function:

$$\text{Loss} = \frac{\alpha \cdot \text{loss}_{\text{VFL}} + \beta \cdot \text{loss}_{\text{GIoU}} + \gamma \cdot \text{loss}_{\text{DFL}}}{\sum_i^{N_{\text{pos}}} \hat{t}}. \quad (5)$$

In all the above formulas, \hat{t} represents the normalized target score. Here, α, β, γ represent the weight coefficient of classification loss, the weight coefficient of regression loss, and the weight coefficient of DFL loss, respectively. The loss_{VFL} indicates the loss of varifocal focus, the $\text{loss}_{\text{GIoU}}$ indicates the GIoU loss, and the loss_{DFL} indicates the loss of distribution focus.

3. Results

In this section, we present the experiment details.

3.1. Experiment Setup. We used a server with a TESLA-V100 GPU for training Fire-PPYOLOE, which has two E5-266v2 CPU and 128 GB of RoM. The operating system is Ubuntu 20.04. Meanwhile, we use a number of libraries of python, such as Paddle, numpy, pycocotools, Cython, pyclicker, PyYAML, and scipy. The number of training rounds is 150 epochs, AdamW [31] is used as the optimizer, and the weight decay is set to 0.0005. The involved optimization strategies are cosine annealing [32] and warning up. The initial learning rate is 0. The learning rate is $1e-4$ at epoch 20, and the final learning rate decays to $1e-6$ at epoch 150. We trained PP-YOLOE with the same settings.

3.2. Forest Fire Dataset. We use a public labeled dataset for the model training and test. The dataset (<https://aistudio.baidu.com/aistudio/datasetdetail/107770>) contains 6,675 fire and smoke images collected on public websites. It is randomly divided into the training set and test set according to the ratio of 80% and 20%. Figure 5 shows two examples of the labeled images. We can see that there are large and small flames in the data, and multiple flames may exist in one image. This demonstrates that the labeled data are close to the real application data.

In real world, the images transmitted from various monitoring devices are unlabeled and in various styles. To evaluate the effect of different models in real scenarios, we also conduct an experiment based on a public unlabeled dataset (<https://www.kaggle.com/datasets/phylake1337/fire-dataset>). It contains 999 images, including 755 images with flames and 244 images without flames. Figure 6 shows two examples of



FIGURE 6: Example of unlabeled images.

unlabeled images. Images with flames are used to detect the recall of the model in the presence of fire, whereas images without fire and smoke are used to detect the false detection rate of the model. From Figure 6, we can see that there are large flames and small flames in the data, and one image may contain one single target or multiple targets. Meanwhile, there are various actual scenes such as images captured in day and night.

3.3. Baseline Models. To verify the effectiveness of our Fire-PPYOLOE, we take Faster R-CNN [33] as a baseline. It is a classical two-stage object detection model with high recall. In this paper, we retrain Faster R-CNN (<https://github.com/rbgirshick/fast-rcnn>) for fair comparison in the same dataset. We take PP-YOLOE (<https://github.com/PaddlePaddle/PaddleDetection/tree/release/2.4/configs/ppyoloe>) as another baseline because it is a state-of-the-art single-stage detection model, and our Fire-PPYOLOE is deployed based on it. We train three models with the optimal training strategy and compare their advantages and disadvantages in this subsection.

3.4. Evaluation Metrics. We compare Fire-PPYOLOE, PP-YOLOE [14], and Faster R-CNN [33] on labeled dataset and unlabeled dataset. For tests on labeled data, we use several commonly used metrics, namely, parameters of different models (Params), giga floating-point operations per second (GFlops), infer time, FPS, and mean average precision (mAP).

Params means the total number of parameters to be trained in our model. Generally speaking, the fewer parameters, the less computation and less memory. GFlops means 1 billion floating-point operations per second, and it is a computational quantity and can be used to measure the complexity of the model. In general, the lower the GFlops, the lower the complexity of the model. Infer time describes the time required to process an image. FPS reflects the number of images that can be processed in 1 s.

The mAP combines a tradeoff between precision and recall, which is a commonly used metric for most detection models. Equations (7) and (8) show the computation formula:

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k, \quad (6)$$

with

$$AP = \sum_{r=0}^1 (r_{n+1} - r_n) P_{\text{interp}}(r_{n+1}), \quad (7)$$

with

$$P_{\text{interp}}(r_{n+1}) = \underset{\tilde{r}: \tilde{r} \geq r_{n+1}}{\sim} \max P(\tilde{r}), \quad (8)$$

where $P(\tilde{r})$ is the measured precision at recall \tilde{r} , and r takes the maximum precision whose recall value is greater or equal than r_{n+1} .

To evaluate the performance of different models on unlabeled dataset, we explore another two metrics in terms of recall and misdetection rate. Specifically, recall means that the correct predictions of positive samples take percentage of all positive samples. The higher value of recall, the better effectiveness, as shown in the following equation:

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (9)$$

where TP represents true positive, which means the number of positive samples that are collectedly detected and FN represents false negative, which means the number of negative samples that are incorrectly detected.

Misdetection means that the negative predictions of negative samples take percentage of all negative samples. The lower number of misdetection, the better effectiveness, as shown in the following equation:

$$\text{Mis-detection} = \frac{FP}{TN + FP}, \quad (10)$$

where FP represents false positive, which means the number of positive samples that are incorrectly detected and TN represents true negative, which means the number of negative samples that are incorrectly detected.

3.5. Evaluation of Models. The Params and GFlops of the Fire-PPYOLOE are much lower than the PP-YOLOE, but the model inference time increases. That was because ConvNeXt use DepthWise convolution and increase the kernel size to replace the small kernel convolution in the original network. DepthWise convolution can reduce parameters effectively, but its memory access time cost is higher than other ordinary convolutions at the same amount of network parameters. For example, replacing ordinary convolutions with deep separable convolutions can reduce the size of network parameters to 10%, but the running speed of the network may only increase 4–5 times. If the DepthWise

TABLE 1: The comparison results of different models on labeled dataset in terms of Params, GFlops, infer time, FPS, and mAP.

Method	Params	GFlops	mAP (%)	Infer time	FPS
PP-YOLOE	50 M	55.5G	37.8	27.7 ms	37
Faster R-CNN	165 M	199G	38.6	72.34 ms	19
Fire-PPYOLOE	37 M	28.5G	43.6	30.80 ms	32

convolution is enlarged to the same size as the ordinary convolution, its running speed will also be much slower than that of the ordinary convolution. This is why ConvNeXt runs much slower compared with ResNet with similar parameter sizes. The purpose of our work is to preserve the high accuracy of ConvNeXt while optimizing its running speed. Our goal is to improve speed as much as possible while ensuring accuracy. Therefore, we add CSPNet network to the model, which can reduce network parameters greatly and restore speed to the same level of PP-YOLOE.

3.5.1. Results on Labeled Dataset. Table 1 shows the results of different models on labeled data. We compare three models in terms of parameters, GFlops, mAP, infer time, and FPS. From Table 1, we draw the following observations:

- (1) From the view of model parameters, Fire-PPYOLOE has only 37 million parameters. This is only 22% of Faster R-CNN and also much smaller than PP-YOLOE. We can see that Fire-PPYOLOE has a relatively low number of parameters and is more suitable for small devices such as drones. This shows the effectiveness of CSPNet, which can decrease the number of parameters largely.
- (2) It shows the same trends in terms of GFlops. The value of Fire-PPYOLOE is 28.5, which is about 50% of PP-YOLOE and only 14% of Faster R-CNN. This shows that Fire-PPYOLOE is much faster. During the experiment, we found that introducing ConvNeXt structure can greatly reduce GFlops.
- (3) For the metric of mAP, we set the IoU to 0.5 and compute the values of different models. We can see that Fire-PPYOLOE is largely superior to the other models. This shows the effectiveness of large kernel convolution. It can perceive a large range of features so as to improve the detection recall.
- (4) As for infer time and FPS, we can see that PP-YOLOE performs the best. Our model performs a little less than PP-YOLOE and much better than Faster R-CNN. We made a tradeoff between the recall and the infer time. Specifically, we combine the ConvNeXt with the PANet. This can further improve the detection precision but a little damage to the infer time.

To sum up, the overall performance of our Fire-PPYOLOE is very good in terms of detection recall and speed. This makes it suitable for practical application in forest fire detection.



FIGURE 7: Picture of successful detection.



FIGURE 8: Picture of misdetection.

3.5.2. Results on Unlabeled Dataset. For the evaluation on unlabeled dataset, we use recall and misdetection rate. Because there is no ground truth for the unlabeled data, we hire volunteers to judge the detection results. For the recall in this test, the judgment is positive if there is fire or smoke in an unlabeled image when it is detected successfully by the model, no matter where the generated box is. For misdetection, if there is no fire or smoke but it is detected by the model incorrectly, the judgement is true. Figure 7 shows an example of successful detection for a given unlabeled forest fire image. Figure 8 shows an example of misdetection picture. In this case, a long yellow road is mistaken for a flame by Fire-PPYOLOE.

Table 2 shows the performances of different models on unlabeled data. We can see that the recall of Fire-PPYOLOE is 81.85%, which is higher than PP-YOLOE through the optimization of large kernel convolution. However, it is relatively low compared with Faster R-CNN. This is because the two-stage detection model has a high advantage in terms of detection accuracy but performs not very well in terms of detection speed, as shown in Table 1.

We can also see that the misdetection rate of Fire-PPYOLOE is only 9.93%, much lower than the other models. By leveraging large kernel convolution, the proposed model can capture a relatively large receptive field, so as to extract more features around the area to be detected. This makes it

TABLE 2: The comparison of different models on unlabeled data in terms of recall and misdetection rate.

	PP-YOLOE (%)	Faster R-CNN (%)	Fire-PPYOLOE (%)
Recall	74.57	97.62	81.85
Misdetection	19.75	19.75	9.93

particularly robust in the detection of some confusing forest fire images. We will make a further qualitative analysis in the following subsection.

To sum up, the proposed Fire-PPYOLOE is more reliable in terms of detection accuracy and speed from the perspective of practical applications. Faster R-CNN has a big disadvantage compared with the other two in terms of both the number of parameters and the detection speed. The original PP-YOLOE is a fast object detector. We improve it using a new designed backbone and neck based on large kernel convolution.

3.6. Qualitative Analysis Results. This subsection gives a qualitative analysis to further evaluate the performances of different models. In practice, there are often smoke- and flame-like objects in forest images, such as the sunset, sunrise, and morning mist. These affect a lot to the performance of forest fire detector. Most research focuses on model optimization such as the improvement of detection recall [34], but few focuses on smoke- and flame-like scenes [13]. Through the experiment, we observe that our Fire-PPYOLOE is able to detect the smoke- and fire-like objects by leveraging large kernel convolution.

YOLO and R-CNN both use backbone to extract features, and neck can fuse multisize feature maps. R-CNN adopts two regressors for classification and regression, respectively, which has high accuracy but slow speed. YOLO adopts one regressor for classification and regression, which has fast speed but low accuracy for small targets. R-CNN is suitable for high-precision detection such as faces and medical images. YOLO is suitable for rapid detection such as autonomous driving and surveillance. Forest fire detection belongs to the monitoring system; therefore, YOLO is preferred. However, it is necessary to improve the initial detection accuracy of flames. Fire-PPYOLOE enhances its ability to distinguish between flames and backgrounds by introducing large kernel convolution to enhance feature maps.

We show three examples of different scenarios, namely, small fire targets, fog in the forest, and fire-like tree trunks. Figure 9 shows the results of PP-YOLOE on the three scenarios. We can see that PP-YOLOE fails to detect small flames. This suggests that the feature extraction for fire is not sufficient by using small kernel convolution.

The results of Faster R-CNN model is shown in Figure 10. It can be seen that Faster R-CNN detect successfully for small fire but incorrectly detect the fogs as smoke and the red tree trunk as fire. This is also because that Faster R-CNN is a two-stage detection model, which focuses excessively on one feature but neglects the extraction of surrounding features, so as to weaken the role of surrounding features as an aid to the central region.

As shown in Figure 11, the proposed Fire-PPYOLOE performs well on three scenarios. It does not misdetect fog as



FIGURE 9: Detection of PP-YOLOE.

smoke nor does it mistake the red tree trunk as fire. It can also detect small fire flame in the forest. However, it does not detect all the flames in the first image. This suggests that the Fire-PPYOLOE is not perfect. There is still room for improvements in terms of recall, as shown in Table 2. It is expected to compensate for the low recall rate in subsequent studies.

By comparing the inspection performance of the models on abundant fire- and smoke-like images, it can be found that Fire-PPYOLOE has better detection efficiency on fire- and smoke-like targets. We use large kernel convolution to feel a larger receptive field, so as to perceive a larger range of feature extraction. Compared with PP-YOLOE and Faster



FIGURE 10: Detection of Faster R-CNN.

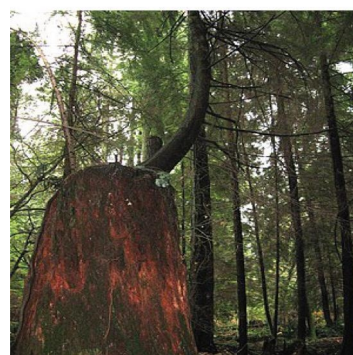
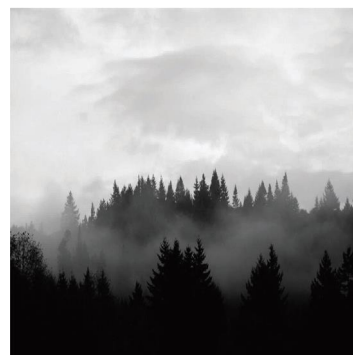


FIGURE 11: Detection of Fire-PPYOLOE.

R-CNN, a larger range of features can better assist judgment, thereby reducing the misdetection rate. Such advantages of the proposed Fire-PPYOLOE can improve the accuracy and reduce the misdetection rate and make it suitable for the practical application as a forest fire detector.

4. Conclusions

In this paper, we propose a new model for the practical application of forest fire detection and discuss its performance compared with state-of-the-art technologies in different scenarios. Based on PP-YOLOE, our model is improved using large kernel convolution to capture surrounding features. It can improve the network precision and reduce the network parameters without too much influence on the reasoning speed. Given a forest image, the proposed model can detect multiple fire and smoke in a time accurately and quickly. This method is not only for early fire detection but also can be applied to other target detection models or

image segmentation models by replacing the infrastructure of other models with CSPConvStage. The paper also puts forward some interesting research directions. For example, it is interesting and necessary to carry out in-depth research on the recognition of small targets.

Data Availability

The online data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by General Project of Science and Technology Plan of Beijing Municipal Education Commission (no. KM202210017006), the 2021–2023 Young Talents

Promotion Project of Beijing Association for Science and Technology, the Beijing Municipal Natural Science Foundation (no. 4214070), National Natural Science Foundation of China (no. 42104175), Teaching Reform project of Beijing Institute of Petrochemical Technology (no. ZDFS GG202103001), Natural Science Foundation of Ningxia (2022AAC03757), 2022 Undergraduate Education and Teaching Reform of China Fire and Rescue Institute (YJYB2022013), and China Fire and Rescue Institute (XFK YB202215).

References

- [1] L. Si, L. Shu, M. Wang et al., "Study on forest fire danger prediction in plateau mountainous forest area," *Natural Hazards Research*, vol. 2, no. 1, pp. 25–32, 2022.
- [2] Y. Bal, B. Wang, Y. D. Wu, and X. D. Liu, "A review of global forest fires in 2021," *Fire Science and Technology*, vol. 41, no. 5, pp. 705–709, 2022.
- [3] L. Ning, L. Qing, X. Jun, and D. Liwen, "Design and implementation of fire detection system based on uav," *Fire Safety Science*, vol. 31, no. 1, Article ID 6, 2022.
- [4] 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 Engineering*, vol. 211, pp. 441–446, 2018.
- [5] X. Wang, T. Li, and L. Z. Wei, "Design and application of forest fire detection system based on image recognition technology," *Fresenius Environmental Bulletin*, vol. 30, no. 11, pp. 11655–11662, 2021.
- [6] W. Mao, W. Wang, Z. Dou, and Y. Li, "Fire recognition based on multi-channel convolutional neural network," *Fire Technology*, vol. 54, pp. 531–554, 2018.
- [7] X. Sun, L. Sun, and Y. Huang, "Forest fire smoke recognition based on convolutional neural network," *Journal of Forestry Research*, vol. 32, pp. 1921–1927, 2021.
- [8] R. Q. Qi and Z. Q. Liu, "Extraction and classification of image features for fire recognition based on convolutional neural network," *Traitement du Signal*, vol. 38, no. 3, pp. 895–902, 2021.
- [9] F. Wu, Z. Yang, X. Mo et al., "Detection and counting of banana bunches by integrating deep learning and classic image-processing algorithms," *Computers and Electronics in Agriculture*, vol. 209, Article ID 107827, 2023.
- [10] Y. Chen, H. Xu, X. Zhang, P. Gao, Z. Xu, and X. Huang, "An object detection method for bayberry trees based on an improved YOLO algorithm," *International Journal of Digital Earth*, vol. 16, no. 1, pp. 781–805, 2023.
- [11] Z. Li, J. Yuan, G. Li et al., "RSI-YOLO: object detection method for remote sensing images based on improved YOLO," *Sensors*, vol. 23, no. 14, Article ID 6414, 2023.
- [12] Z. Jiao, Y. Zhang, L. Mu et al., "A yolov3-based learning strategy for real-time uav-based forest fire detection," in *2020 Chinese Control and Decision Conference (CCDC 2020)*, pp. 4963–4967, IEEE, Hefei, China, August 2020.
- [13] 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, Article ID 4930, 2022.
- [14] S. Xu, X. Wang, W. Lv et al., "Pp-yoloe: an evolved version of yolo," 2022.
- [15] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9627–9636, IEEE, New Orleans, LA, USA, 2019.
- [16] X. Huang, X. Wang, W. Lv et al., "PP-YOLOv2: a practical object detector," 2021.
- [17] C. Feng, Y. Zhong, Y. Gao, M. R. Scott, and W. Huang, "TOOD: Task-aligned one-stage object detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3490–3499, IEEE Computer Society, 2021.
- [18] S.-C. Pei and C.-N. Lin, "Image normalization for pattern recognition," *Image and Vision Computing*, vol. 13, no. 10, pp. 711–723, 1995.
- [19] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11976–11986, IEEE, New Orleans, LA, USA, June 2022.
- [20] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: a new backbone that can enhance learning capability of cnn," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 390–391, IEEE, Seattle, WA, USA, June 2020.
- [21] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8759–8768, IEEE Computer Society, Los Alamitos, CA, USA, 2018.
- [22] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 29, 2016.
- [23] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31x31: revisiting large kernel design in CNNs," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11963–11975, IEEE, New Orleans, LA, USA, June 2022.
- [24] J. Li, C. Wang, B. Huang, and Z. Zhou, "ConvNeXt-backbone HoVerNet for nuclei segmentation and classification," 2022.
- [25] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.
- [26] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *ICML'15: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, pp. 448–456, PMLR, Lille, France, 2015.
- [27] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016.
- [28] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017.
- [29] H. Zhang, Y. Wang, F. Dayoub, and N. Sünderhauf, "VarifocalNet: an IoU-aware dense object detector," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8514–8523, IEEE, Nashville, TN, USA, June 2021.
- [30] X. Li, W. Wang, L. Wu et al., "Generalized focal loss: learning qualified and distributed bounding boxes for dense object detection," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21002–21012, 2020.
- [31] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017.
- [32] I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts," 2016.
- [33] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, IEEE, Santiago, Chile, December 2015.
- [34] X. Zhang, K. Qian, K. Jing, J. Yang, and H. Yu, "Fire detection based on convolutional neural networks with channel attention," in *2020 Chinese Automation Congress (CAC)*, pp. 3080–3085, IEEE, Shanghai, China, November 2020.