Fire Ecology

ORIGINAL RESEARCH

Open Access

Real-time fire detection algorithms running on small embedded devices based on MobileNetV3 and YOLOv4

Hongtao Zheng[1†], Junchen Duan[1†], Yu Dong[2] and Yan Liu[1*]

Abstract

Aim Fires are a serious threat to people's lives and property. Detecting fires quickly and effectively and extinguishing them in the nascent stage is an effective way to reduce fire hazards. Currently, deep learning-based fire detection algorithms are usually deployed on the PC side.

Methods After migrating to small embedded devices, the accuracy and speed of recognition are degraded due to the lack of computing power. In this paper, we propose a real-time fire detection algorithm based on MobileNetV3-large and yolov4, replacing CSP Darknet53 in yolov4 with MobileNetV3-large to achieve the initial extraction of flame and smoke features while greatly reducing the computational effort of the network structure. A path connecting PANet was explored on Gbneck(104, 104, 24), while SPP was embedded in the path from MobileNetV3 to PANet to improve the feature extraction capability for small targets; the PANet in yolo4 was improved by combining the BiFPN path fusion method, and the improved PANet further improved the feature extraction capability; the Vision Transformer model is added to the backbone feature extraction network and PANet of the YOLOv4 model to give full play to the model's multi-headed attention mechanism for pre-processing image features; adding ECA Net to the head network of yolo4 improves the overall recognition performance of the network.

Result These algorithms run well on PC and reach 95.14% recognition accuracy on the public dataset BoWFire. Finally, these algorithms were migrated to the Jeston Xavier NX platform, and the entire network was quantized and accelerated with the TensorRT algorithm. With the image propagation function of the fire robot, the overall recognition frame rate can reach about 26.13 with high real-time performance while maintaining a high recognition accuracy.

Conclusion Several comparative experiments have also validated the effectiveness of this paper's improvements to the YOLOv4 algorithm and the superiority of these structures. With the effective integration of these components, the algorithm shows high accuracy and real-time performance.

Keywords Fire detection, YOLOv4, MobileNetV3-large, PANet, BiFPN, SPP, ECA Net, TensorRT

[†] Hongtao Zheng and Junchen Duan are the co-first authors.

*Correspondence:
Yan Liu
liuy0808@126.com
[1] School of Information and Electrical Engineering, Zhejiang University City College, Hangzhou 310015, China
[2] School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China

Resumen

Antecedentes Los incendios representan una seria amenaza para la gente y sus propiedades. El detectar incendios rápida y efectivamente y extinguirlos en su estado inicial es una forma efectiva de reducir sus peligros. En la actualidad, la detección de incendios basada en algoritmos de detección usando el conocimiento profundo (deep learning) están siendo desarrollados mediante el uso de computadores (PCs).

Metodología Luego de migrar hacia computadores cada vez más pequeños, la exactitud y velocidad de reconocimiento se están degradando debido a una falta de capacidad de computación. En este trabajo, proponemos un algoritmo de detección de incendios en tiempo real basado en la tecnología digital yolov4, que reemplaza a CSP Darknet53 en el yolov4 por la MobileNetV3-large, para alcanzar las características iniciales que permitan la detección de llamas y humo mientras se reduce grandemente el esfuerzo en la estructura de las redes computacionales; un paso que conecta PANet fue explorado con Gbneck(104,104,24), mientras que SPP fue incorporado en el paso que conecta MobileNetV3-large a PANet para mejorar la capacidad de extracción sobre las características de objetivos pequeños. El PANet en el yolov4 fue mejorado combinando el método de fusión BiFPN, y este PANet mejorado incrementó además las características en la capacidad de extracción. El modelo de Vision Transformer es adicionado a la columna vertebral de las características de la red de extracción y al modelo yolov4 para brindar una mayor articulación al mecanismo de atención del modelo de cabezas múltiples para pre-procesar las características de las imágenes. La adición de la RED ECA a la cabeza de la red yolov4 mejora la performance del reconocimiento general de la red.

Resultados Estos algoritmos funcionan bien en una PC y alcanzan y reconocen una exactitud del 95% en el conjunto de datos públicos BoWFire. Finalmente, estos algoritmos fueron migrados a la plataforma Jeston Xavier NK y la red completa fue cuantificada y acelerada con el algoritmo Tensor RT. Con la función de propagación de imagen del robot de fuego, el reconocimiento general de la tasa de encuadre puede alcanzar 26.13, con una performance en tiempo real mientras se mantiene una alta exactitud de reconocimiento.

Conclusiones Diferentes experimentos comparativos han validado también la efectividad de este trabajo en el mejoramiento del algoritmo yoylov4 y la superioridad de estas estructuras. Con la interacción efectiva de estos componentes, el algoritmo muestra una alta exactitud y performance en tiempo real.

Introduction

Fire is one of the major public safety disasters that can result in casualties and economic and property losses. Detecting fire conditions as early as possible and extinguishing them in the beginning stages is an effective method to reduce fire hazards. Therefore, researching rapid and accurate fire detection is of great significance (Muhammad et al. 2018a, b, c). Traditional smoke detectors can sense fire when smoke particles enter a room, but this method has a long detection time and is not suitable for outdoor fire detection.

With the development of neural networks and deep learning and other fields (Gong et al. 2021; Succetti et al. 2022), a video-based fire detection method is proposed. Compared with traditional methods, it has the advantages of fast response, non-contact, visualization, intelligence, and easy integration. Most fires pass through a long-smoldering process before the occurrence of a flame, generating a large amount of smoke. Due to the diffusion of smoke, smoke can identify the trend of fire earlier than flame detection, and the response time is earlier.

Although the smoke detection algorithm has made great progress, it has not been widely used in the real world, mainly because of the following reasons: fire generally causes the background scene to become complicated, thereby reducing the accuracy of the detection algorithm, false alarms, leaks fire alarms, and other phenomena occur frequently; although the general fire detection algorithm has good accuracy, it is too complicated, which will cause it to not run well on general small embedded devices. If the algorithm does not run stably on some embedded platforms, then such algorithms lose their practical applicability.

Based on the above analysis, we conclude that the limitations of current fire detection algorithms include too many parameters for the algorithm to calculate and poor immunity to environmental disturbances resulting in the algorithm being prone to false alarms. For these reasons, in this paper, we propose a new lightweight fire detection algorithm. The contribution of the algorithm is as follows:

1. It is proposed to replace the backbone network CSP-Draknet53 of YOLOv4 (Bochkovskiy et al. 2020) with the MobileNetV3 (Howard et al. 2019) network, which can effectively extract valid information and greatly reduce the computational complexity of the algorithm.
2. In this paper, the YOLOv4 algorithm improves multiscale feature fusion by extending a PANet (Liu et al. 2018) path at the G-bneck (104, 104, 24) layer to improve the detection of multi-pose and multi-scale targets.
3. The Spatial Pyramid Pooling (SPP (He et al. 2015)) module is added to the path from the feature layer of the backbone output to the PANet to improve the feature extraction of small targets.
4. The path fusion method based on BiFPN (Tan et al. 2020) is used to improve the path aggregation method of PANet to further improve the feature extraction capability.
5. The Vision Transformer (Dosovitskiy et al. 2020) model is added to the backbone feature extraction network and PANet of the YOLOv4 model to give full play to the model's multi-headed attention mechanism for pre-processing image features.
6. Efficient Channel Attention (ECA) (Wang et al. 2020) is added to the header network of YOLOv4, which reduces the input of interference information and improves the overall recognition effect of the network.
7. The algorithm running stably on PC was successfully migrated to Jeston Xavier NX, and TensorRT was used to accelerate the algorithm.
8. For the model training and experimental comparison of this algorithm, we collected a series of flame and smoke images, including single flame and smoke, multi-body flame and smoke, indoor fire, forest fire, and complex background fire scenarios, with a total of 29,980 images, divided into a training set, a validation set and a test set according to a ratio of 7:1:2.

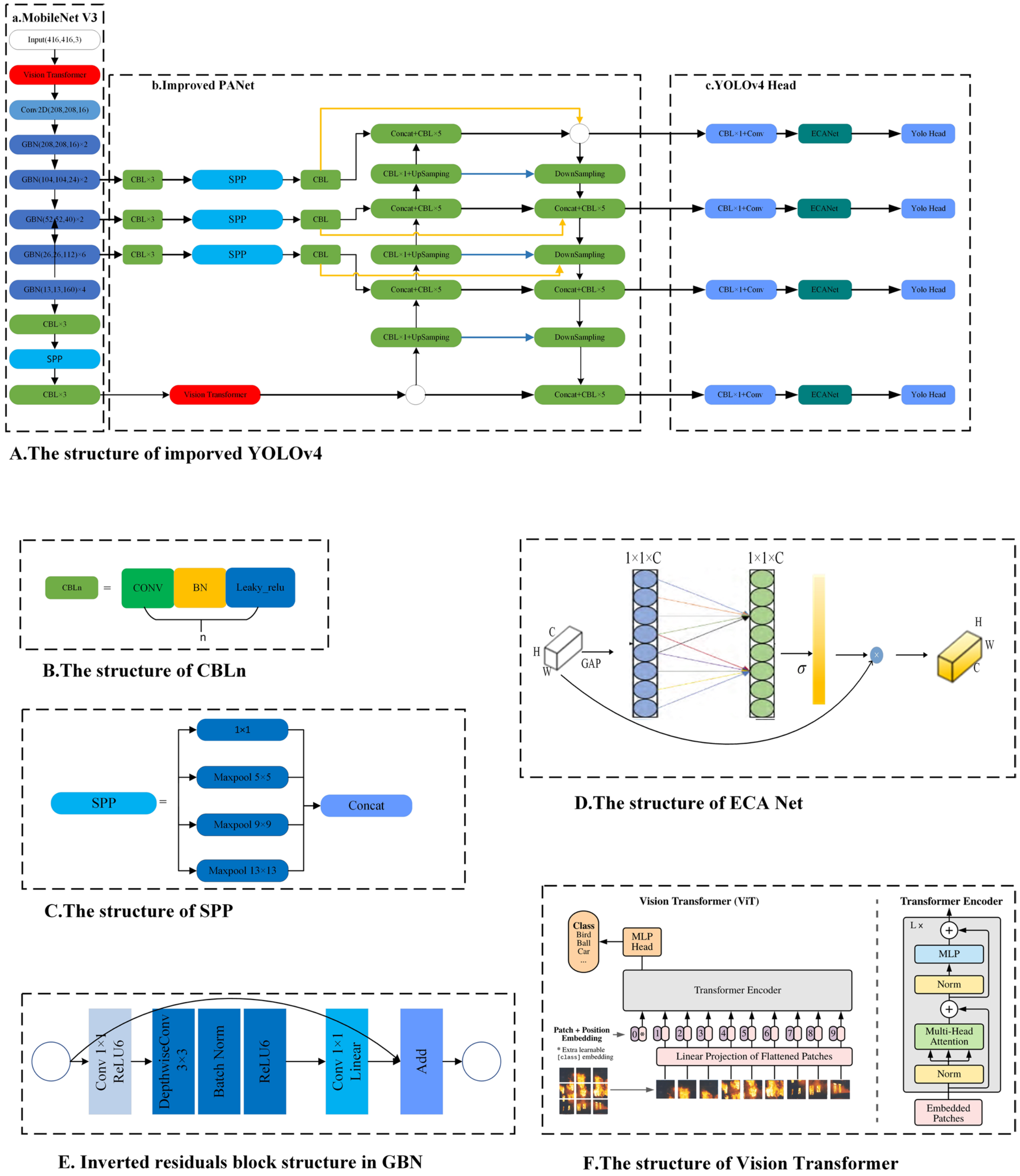
Related work

Traditional fire detection is typically based on a combination of flame and smoke sensors, but this type of method has severe restrictions on the environment used and cannot be used in all situations. With the widespread use of video cameras in public safety systems, fire detection techniques based on machine learning methods of image information have been rapidly developed. Traditional vision-based fire detection methods generally achieve fire detection by extracting fire features, such as color (Töreyin et al. 2006; Chen et al. 2006; Genovese et al. 2011, Celik and Demirel 2009), texture (Gunay et al. 2012; Chunyu et al. 2009; Yuan et al. 2016a, b; Dimitropoulos et al. 2016), shape (Hongyu et al. 2020; Töreyin et al. 2005), and motion state (Han and Lee 2009, Yuan 2008). Related research results are as follows: Kim et al. (2014) established an RGB color model to achieve fire detection, but the robustness and generalization ability of the method was insufficient; Wang et al. (2020) proposed a fusion of flame color and local features, a flame detection method based on KNN background subtraction; Günay and Çetin (2015) proposed a real-time dynamic texture recognition method using projection to random hyperplanes and deep neural network filters and applied the method to infrared video, real-time flame detection. Emmy Prema et al. (2018) preliminarily segmented the flame regions in the image according to the YCbCr color space and extracted static and dynamic texture features for the candidate flame regions through 2D 1446 Fire Technology 2022 temporal wavelet decomposition and 3D volume wavelet decomposition. Finally, the candidate flame regions are classified according to the extracted texture features. Jia et al. (2016) adopted non-linear enhanced smoke color features to identify smoke regions, then used motion features to measure saliency, and finally used motion energy and saliency maps to segment smoke regions. Habiboğlu et al. (2012) divided the video into spatiotemporal blocks and used the covariance-based spatiotemporal features extracted from these blocks to train an SVM classifier. Dimitropoulos et al. (2014) employed background subtraction and color analysis to define candidate regions, and then modeled fire behavior in time and space using color probability, flicker, space, and energy simultaneously for each candidate region, and performed dynamic texture analysis. Finally, the candidate regions are classified using a two-class SVM classifier. Yuan et al. (2016a, b) proposed a method for forest fire detection using drones. Firstly, the candidate area is extracted by the color feature of the flame; then, the motion vector of the candidate area is calculated by the Horn-Schunck optical flow algorithm, and the binary image is obtained by thresholding and morphological operation on the motion vector. Finally, the spot counting method is used to locate the fire source in the binary image. Kim and Lattimer (2015) and Kim et al. (2016) extracted the texture and motion features of flames and smoke from long-wave infrared images for autonomous navigation of robots in fire environments. Although these algorithms are less dependent on the computing power of the hardware, the accuracy of detection is affected by the accuracy of the algorithm's feature extraction and is also susceptible to interference from the environment, and the shape and color characteristics of flames and smoke are very complex and variable. It is becoming clear that traditional vision algorithms alone cannot effectively solve these problems.

The subsequent rise of fields such as neural networks, artificial intelligence, and deep learning has provided new opportunities to address fire detection. Related research results are as follows: Frizzi et al. (2016) used a 6-layer CNN to solve the three classification problems of fire, smoke, and no fire. Tao et al. (2016) used deep convolutional neural networks to achieve end-to-end training from raw pixel values to classifier output, which successfully improved the accuracy of smoke detection. Yin et al. (2017) proposed a 14-layer deep normalized convolutional neural network (DNCNN) to achieve automatic extraction of smoke features. Xu et al. (2021) applied deep learning techniques to adaptively learn and extract the features of forest fires. The method first integrated two independent learners, Yolov5 and EfficientDet, to complete the fire detection process. Second, another individual learner, EfficientNet, is responsible for learning global information to avoid false positives, and finally, the detection results are based on the decisions of the three learners. Kim and Lee (2019) proposed a deep learning-based fire detection method using video sequences, which uses a convolutional neural network (R-CNN) to detect suspicious fire areas (SRoF) and non-hazardous fires based on their spatial features. fire area. Then, the aggregated features within bounding boxes in consecutive frames are accumulated by LSTM to classify whether there is a fire in the short term. Decisions in successive short periods are then combined into a majority vote for the final decision in the long period. Zhang et al. (2018) solved the problem of insufficient training data by inserting real smoke or simulated smoke into the forest background to generate synthetic smoke images and used the synthetic smoke image dataset to train Faster R-CNN to obtain a smoke detection model. Xu et al. (2019) proposed a novel deep saliency network-based method for video smoke detection. Informative smoke saliency maps are extracted by combining pixel-level saliency convolutional neural networks and object-level saliency convolutional neural networks, and the presence of smoke in images is predicted by combining deep feature maps and saliency maps. Lin et al. (2019) constructed a joint framework of RCNN and 3D CNN, using RCNN to extract static spatial information and using 3D CNN to extract spatiotemporal features, thus solving the problem of fire smoke detection and localization. It can be seen that the complex CNN can extract the spatial features of the smoke target and can accurately locate the smoke in time, which is very suitable for smoke detection. Bhattarai and Martinez-Ramon (2020) used deep convolutional neural networks to extract, process, and analyze key information from thermal imaging, creating an automated system capable of detecting critical objects at fire sites in real time. Wu et al. (2022) proposed a video fire detection algorithm based on YOLOv5, which improved SPP, and used an activation function (GELU) and predictive bounding box suppression (DIoU-NMS), with excellent performance of the final algorithm. Huang et al. (2023) proposed a light forest fire detection algorithm with a defogging function. The algorithm first obtains a fog-free image after a dark channel operation on the image and then detects the image with a lightened and improved YOLO-L-Light algorithm. Xue et al. (2022) proposed an improved forest fire classification and detection algorithm based on YOLOv5, which introduced SIoU and CBAM, and improved PANet to a BiFPN-like structure, and the final algorithm outperformed the original algorithm in all aspects. Zhao et al. (2022) proposed an improved YOLO algorithm that extends the feature extraction network in three dimensions and adds feature propagation properties to improve the network performance and reduce the algorithm parameters. Sathishkumar et al. (2023) proposed a learning without forgetting (LwF) method for fire detection algorithms, which addresses the possibility that the detection model may lose its ability to classify the original dataset when applying migration learning thereby greatly reducing the number of steps required to migrate the detection model for learning. Zheng et al. (2023) novel algorithm for remote sensing forest fire detection is proposed, which first uses FireYOLO for the initial recognition of the target, then applies the Real-ESRGAN algorithm to the target to improve image clarity, followed by FireYOLO for a second recognition. Each of these algorithms has its own characteristics and solves some of the challenges in fire detection, but it is still a challenging problem to improve the accuracy of the algorithm and its immunity to interference while reducing the number of parameters to a great extent.

Materials and methods

In order to further improve the real-time performance of the deep learning-based fire detection algorithm, this paper proposes a fire detection algorithm based on MobileNetV3-large and YOLOv4 (hereafter referred to as MobileNetV3-large-YOLOv4 algorithm), the structure



A.The structure of imporved YOLOv4

B.The structure of CBLn

C.The structure of SPP

D.The structure of ECA Net

E. Inverted residuals block structure in GBN

F.The structure of Vision Transformer

Fig. 1 The network structure of MobileNetV3-large-YOLOv4

of which is shown in Fig. 1. The algorithm improves the network structure of YOLOV4: the MobileNetV3-large is used as the backbone network to achieve the initial extraction of smoke and flame features; the PANet path is extended at the G-bneck(104, 104, 24) layer to improve the multi-scale feature fusion and enhance the detection of multi-pose and multi-scale targets; the feature layer at the backbone output to the PANet's path, the SPP module, is added to improve the feature extraction of small targets; the path of PANet is modified according to the path connection principle of BiFPN; the Vision Transformer model is added to the backbone feature extraction

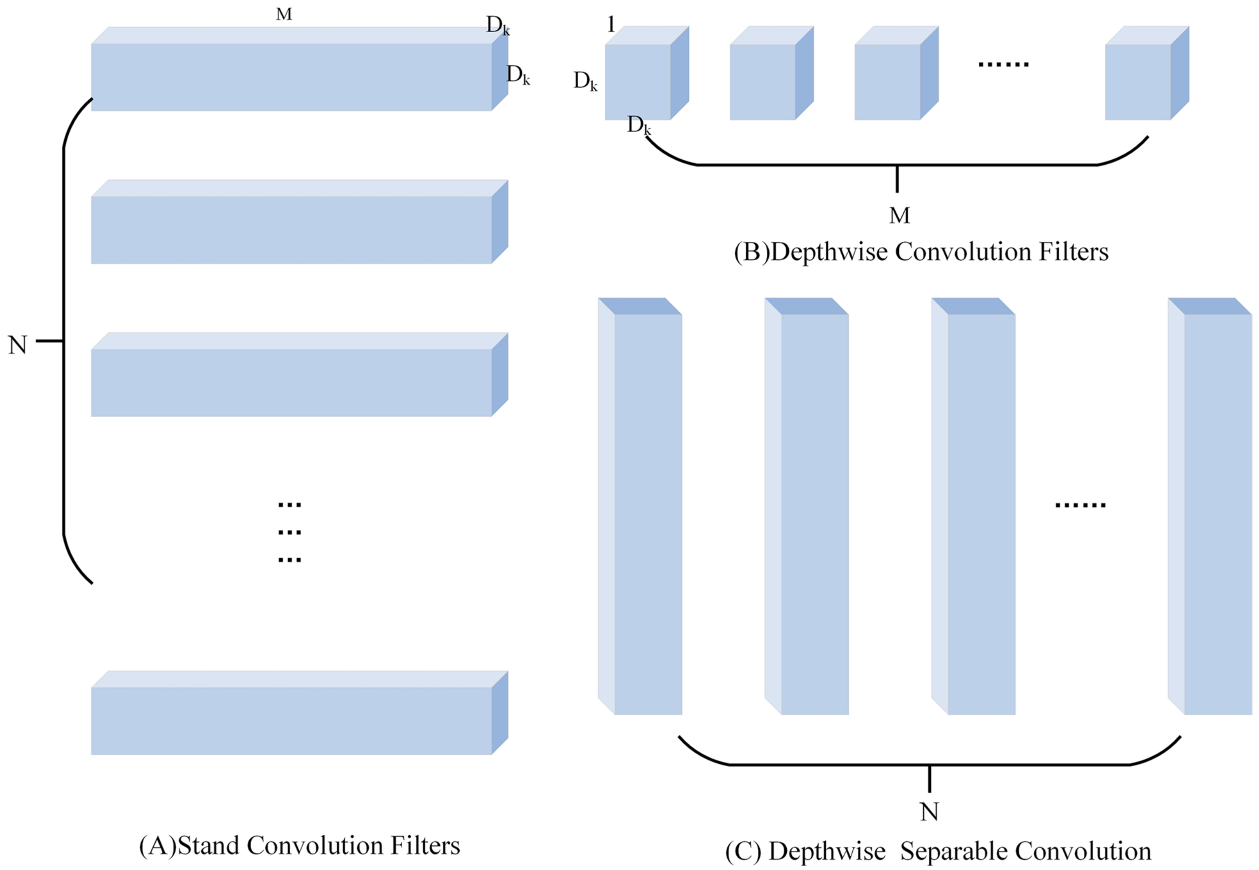

M
D_k
D_k
N
...
...
...
(A)Stand Convolution Filters
1
D_k
D_k
D_k
......
M
(B)Depthwise Convolution Filters
......
N
(C) Depthwise Separable Convolution

Fig. 2 Depth-separable convolution structure

network of yolo4 model and PANet to give full play to the multi-head attention mechanism of the model to pre-process the image features; the ECANet is introduced in the head network to reduce the input of interference information and improve the extraction of effective information. The algorithm runs well on PC and achieves a recognition accuracy of 95.04% on the public dataset BoWFire (Chino et al. 2015). Finally, these algorithms are migrated to the Jeston Xavier NX platform to quantify and accelerate the entire network using the TensorRT algorithm. Using the image propagation function of the fire robot, the overall recognition frame rate can reach about 26.13, and the algorithm has a high real-time performance while maintaining a high recognition accuracy.

Fire feature extraction based on MobileNetV3-large

The MobileNet network is a lightweight CNN proposed by Google. The convolution model of MobileNetV1 mainly uses the depthwise separable convolution (depthwise separable convolution) to replace the ordinary convolution method. The depthwise separable convolution process is shown in Fig. 2. It is achieved by using different convolution kernels for each input channel to perform convolution, and then channel adjustment through 1×1 convolution kernel, and add a BN (Batch Normalization) layer and ReLU after the convolution layer, activation function. Suppose the size of the input feature map is $D_W \times D_H \times M$, and the size of the output feature map is $D_W \times D_H \times N$, where D_W and D_H are the width and height of the feature map, respectively, and M and N are the number of channels of the input and output feature maps, respectively. For a standard convolution with a convolution kernel size of $D_K \times D_K$, there are N convolution kernels of $D_K \times D_K \times M$, so the calculation formula of the parameter PN can be expressed as:

$$P_N = D_K \times D_K \times M \times N \tag{1}$$

Each convolution kernel has to undergo $D_W \times D_H$ calculations, and its calculation amount S_N table is shown as:

$$S_N = D_K \times D_K \times M \times N \times D_W \times D_H \tag{2}$$

In depthwise separable convolution, a standard convolution can be divided into depthwise convolution and pointwise convolution two-step operation. Depthwise convolution requires only a $D_K \times D_K \times M$ Convolution kernel; the size of the convolution kernel of pointwise convolution is $1 \times 1 \times M$, and there are N in total, because this parameter P_D is expressed as:

$$P_D = D_K \times D_K \times M + M \times N \tag{3}$$

Each parameter of depthwise convolution and pointwise convolution needs to go through $D_W \times D_H$ operations, and its computational cost S_D is expressed as:

$$S_D = D_K \times D_K \times M \times N \times D_W \times D_H + M \times N \times D_W \times D_H \tag{4}$$

The ratio of depthwise separable convolution modules to standard convolution parameter quantities R_P table. It is shown as formula (5), and the calculation ratio R_Q is expressed as formula (6).

$$R_P = \frac{P_D}{P_N} = \frac{1}{N} + \frac{1}{D_K^2} \tag{5}$$

$$R_Q = \frac{S_D}{S_N} = \frac{1}{N} + \frac{1}{D_K^2} \tag{6}$$

It can be seen from the above formula that the parameters and calculation amount of the depthwise separable convolution are reduced $\frac{1}{N} + \frac{1}{D_K^2}$ for standard convolution.

The MobileNetV1 network structure is prone to failure of the convolution kernel of the depth convolution part during the training process, that is, most of the parameters of the convolution kernel are 0, which affects the feature extraction effect. MobileNetV2 uses the inverted residuals block (Sandler et al. 2018) structure on the basis of V1, as shown in Fig.1E. Firstly, point-by-point convolution is used to increase feature dimension, then depthwise convolution is used for feature extraction, and finally point-by-point convolution is used for dimension reduction, and the ReLU activation function is replaced with the ReLU6 activation function, which makes the model more powerful under low-precision computing robustness and remove the last ReLU layer. The formula of the ReLU6 activation function is expressed as:

$$\mathrm{Re}LU6(x) = \min(\max(0, x), 6) \tag{7}$$

When the input dimension is the same as the output dimension, the residual connection in ResNet is introduced to directly connect the output with the input. The characteristics of this inverted residual structure are that the upper and lower layers have low feature dimensions, and the middle layer has high dimensions, which avoids the failure of the convolution kernel in the deep convolution process of MobileNetV1, and the use of single depth convolution in the high-dimensional feature layer is not would increase the amount of parameters too much. In addition, the introduction of residual connections can avoid the phenomenon of gradient disappearance when deepening the network depth.

MobileNetV3 uses a 3×3 standard convolution and multiple bneck structures to extract features. After the feature extraction layer, a 1×1 convolution block is used to replace the fully connected layer, and a maximum pooling layer is added to obtain the final classification result, which further reduces the amount of network parameters. MobileNetV3 includes two structures, large and small, and this paper uses the large structure. In order to adapt to the model recognition task, the input image size is set to 416×416. The structure of MobileNetV3_large is shown in Table 1, where SE means whether to use the attention module, NL means which activation function to use, and s means the step size.

Improvement and optimization of the neck structure

The PANet structure is used in four feature layers

The shallow network contains more localization information, while the deep network contains more semantic information, and the localization information of small-scale pedestrians is lost after a series of downsampling operations. The aim of this section is to improve the detection accuracy of the YOLOv4 detection model for small-scale flames and smoke by fusing multi-scale features so that more localization information of the shallow small target line flames and smoke is transferred to the deeper network.

The PANet structure first performs top-to-bottom feature extraction in the traditional feature pyramid structure FPN (Lin et al. 2017) (Feature Pyramid Network), which only enhances semantic information and does not convey localization information, then completes bottom-to-top path-enhanced feature extraction in the next feature pyramid, which conveys strong localization information in the shallow layer; next, the adaptive feature pooling layer uses features from each layer of the pyramid to enable more accurate classification and localization at a later stage. The next layer is the adaptive feature pool layer, which uses features from each layer of the pyramid to enable more accurate classification and localization at a later stage. Figure 3 shows various types of relational network structures related to the neck structure of this paper, where a is FPN, b is PANet, c is BiFPN, and d is the neck relational network structure of the algorithm in this paper, which is obtained by fusing the structural features of PANet and BiFPN.

The YOLOv4 algorithm uses the PANet structure on the three effective feature layers, but it is still not

Table 1 MobileNetV3-large construction method diagram

Input	Operator	#exp	Out	SE	NL	Stride
$416^2 \times 3$	Conv2d	–	16	–	HS	2
$208^2 \times 16$	bneck	16	16	–	RE	1
$208^2 \times 16$	bneck	64	24	–	RE	2
$104^2 \times 24$	bneck	72	24	–	RE	1
$104^2 \times 24$	bneck	72	40	1	RE	2
$52^2 \times 40$	bneck	120	40	1	RE	1
$52^2 \times 40$	bneck	120	40	1	RE	1
$52^2 \times 40$	bneck	240	80	–	HS	2
$26^2 \times 80$	bneck	200	80	–	HS	1
$26^2 \times 80$	bneck	184	80	–	HS	1
$26^2 \times 80$	bneck	184	80	–	HS	1
$26^2 \times 80$	bneck	480	112	1	HS	1
$26^2 \times 112$	bneck	672	112	1	HS	1
$26^2 \times 112$	bneck	960	160	1	HS	2
$13^2 \times 160$	bneck	960	160	1	HS	1
$13^2 \times 160$	bneck	960	160	1	HS	1
$13^2 \times 160$	Conv2d	–	960	–	HS	1
$13^2 \times 960$	Pool	–	–	–		1
$1^2 \times 960$	Conv2d	–	1280	–	HS	1
$1^2 \times 1280$	Conv2d	–	1000	–		1

effective in recognizing small target pedestrians and multi-attitude pedestrians. Therefore, YOLOv4 is improved as shown in Fig. 3 to perform multi-scale feature fusion on the four effective layers.

Medium- and large-scale feature layers introduce SPP structure

The SPP structure was originally used as a transition layer between convolutional layers and fully connected layers to solve the problem of size mismatch. Subsequently, researchers found that this structure can enhance the receptive field; therefore, some researchers tried to introduce the improved SPP module into the target detection network, splicing multi-scale local area features, and improving the accuracy of target detection (Huang et al. 2020; Mao et al. 2020). YOLO V3-608 with SPP module outperforms AP50 by 2.7% in the COCO object detection task.

In YOLO V4, the SPP module is set after the small-scale feature layer to be responsible for the prediction of small and medium-sized targets, but the SPP module is not set after the medium-scale feature layer and the large-scale feature layer. Therefore, it is easy to lose small objects during the propagation process.

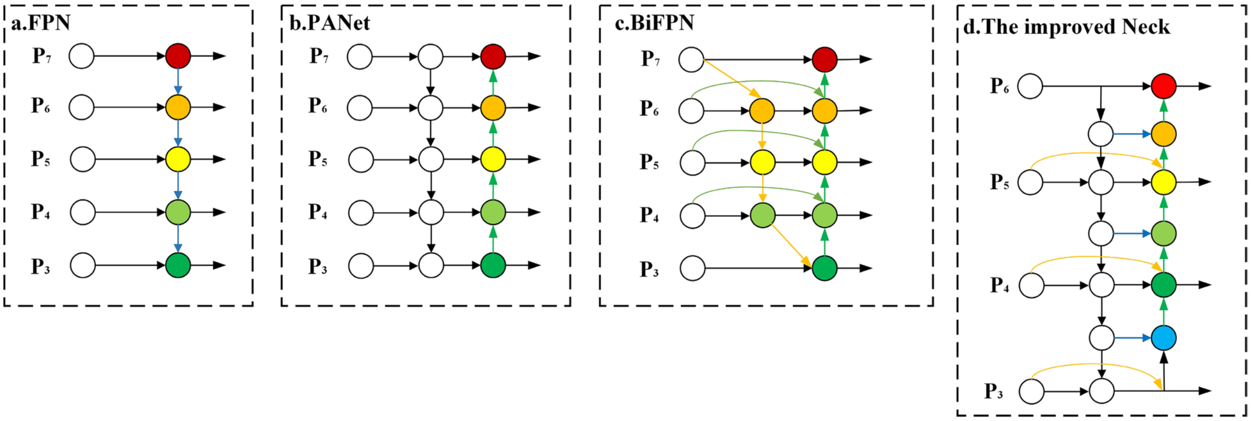


Fig. 3 Schematic representation of the various types of network relationship structures

The characteristics of the target lead to the omission of small targets. In this paper, after the medium-scale feature layer and the large-scale feature layer, the SPP module is added, and the feature tensors of different scales extracted by the backbone network are input into the SPP module, so that the characteristics of small and medium-sized targets are more obvious, target identification.

The SPP module consists of three max-pooling layers and one connection layer. Figure 1 (c) shows the structure of the SPP module. The maximum pooling is performed by a pooling core, the size of the pooling core is 5×5, 9×9, and 13×13, and the step length of the pooling core is 1. Therefore, after the pooling operation, three new feature maps of the same size as the original feature map are obtained. The three feature maps are superimposed with the original feature maps to get the final output of the module.

Improved PANet

When three feature tensors of different scales pass through the SPP module, the PANet structure is adopted for further feature fusion. Small-scale features are more responsive to the overall target, while large-scale features are better at expressing local features. However, considering the use of MobileNetV3-large to replace CSPDarknet53, YOLOv4 is lightweight and reduces the ability of feature fusion. This paper applies the path fusion idea in BiFPN to improve PANet in YOLOv4. In BiFPN, the input nodes and output nodes of the same layer can be connected across layers to ensure that more features are incorporated without increasing the loss. This algorithm performs cross-layer connections on the same level of PANet (the three orange lines in Figs. 1A and 3d); in this way, the path from low-level information to high-level information can be shortened, and their semantic features can be combined. In BiFPN, adjacent layers can be merged in series. In this paper, the adjacent layers of PANet are merged in series (the three blue lines in Figs. 1A and 3d).

The improved PANet has the characteristics of bidirectional cross-scale connection and weighted feature fusion, which improves the feature fusion ability and further increases the feature extraction ability.

Introduction of ECA attention mechanism

When learning and understanding the unknown, humans quickly focus their attention on key areas and ignore useless information in order to get the information they need quickly and accurately. Researchers have been inspired to incorporate attention mechanisms into convolutional neural networks to improve the performance of traditional network models while sacrificing a small amount of computation. In this paper, the Efficient Channel Attention (ECA) module is added to YOLOv4, and the weights are trained on the channel dimensions of the four feature layers of the head network to make the model more focused on useful information. The specific structure of ECA Net is shown in Fig. 1D.

The ECA module can be seen as an improved version of the Squeeze-and-Excitation (SE (Hu et al. 2018)) module. The authors of the ECA argue that the SE prediction of the channel attention mechanism has the side effect of capturing all channel dependencies inefficiently and unnecessarily, whereas convolution has good cross-channel information acquisition capabilities, so the ECA module replaces the 2-full joins of the SE module with 1D convolution. The size of the convolution kernel of the 1D convolution affects the coverage of cross-channel interactions, so it is important to choose the 1D convolution kernel size *k*. Although *k* can be adjusted manually, this wastes a lot of time and effort. *k* is non-linearly proportional to *C*. The larger *C* is, the stronger the long-term interaction; conversely, the smaller *C* is, the stronger the short-term interaction, i.e.:

$$C = \emptyset(k) = 2^{(\gamma \times k - b)} \tag{8}$$

Once the channel dimension *C* has been determined, the convolution kernel size *k* is then:

$$k = \varphi(C) = \left| \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right| \tag{9}$$

where γ and b are the regulation parameters; $|t|_{\text{odd}}$ denotes the nearest odd number *t*.

In this paper, the ECA module is applied to the enhanced feature extraction network by adding an attention mechanism to the 152×152, 76×76, and 38×38 feature layers extracted from the backbone network, so that the subsequent training of the network can focus on the effective features and improve the detection capability of the algorithm.

Introduction of Vision Transformer

The Vision Transformer model was designed with the principle of not changing the transformer too much, using the Transformer Encoder part to do the classification, i.e., just to solve the problem of its poor performance in classification tasks with large data. Alexey Dosovitskiy et al. were inspired by the success of transformer scaling in NLP and attempted to apply the standard transformer directly to images with as few modifications as possible, eventually proposing the Vision Transformer for computer vision modules, whose network structure is shown in Fig. 1F.

Table 2 The number of each type of dataset

Types of datasets	Total number
Training sets	21,009
Test sets	2973
Validation set	5998

The Vision Transformer first chunks the image and then adds a classification token to the image sequence so that the sequence of images is cut into smaller chunks from a single image, with the dimensionality changing as shown in Eq. 10.

$$B, C, H, W \Rightarrow B, N, P^2CN = \frac{HW}{P^2} \tag{10}$$

Instead of using the traditional transformer encoding method, the Vision Transformer's position encoding first initializes the position information randomly and then trains to learn the image features. Finally, the extracted image features are used to generate feature predictions for different target classes. The encoding used is shown in Eqs. 11 and 12.

$$\mathrm{PE}(\mathrm{pos}, 2i) = \sin(\frac{\mathrm{pos}}{10,000^{\frac{2i}{d_{\mathrm{model}}}}}) \tag{11}$$

$$\mathrm{PE}(\mathrm{pos}, 2i+1) = \cos(\frac{\mathrm{pos}}{10,000^{\frac{2i}{d_{\mathrm{model}}}}}) \tag{12}$$

Algorithm quantization based on TensorRT

To have a faster operation speed on the embedded platform, this paper further quantifies the related algorithms. The commonly used methods are network pruning, model quantization, and so on. Considering that the MobileNetV3-large-YOLOv4 algorithm has adopted the MobileNetV3-large lightweight network structure, continuing to prune the MobileNetV3-large-YOLOv4 network will destroy the integrity of the entire network, so this paper adopts the model quantization method to achieve the quantization of the algorithm.

Table 3 The anchor of fire and smoke

Size			
The flame's prior box size (w,h)			
13×13	(116,220)	(155,230)	(367,67)
26×26	(41,72)	(71,55)	(61,105)
52×52	(18,31)	(24,81)	(27,67)
The smoke's prior box size (w,h)			
13×13	(96,210)	(115,170)	(263,43)
26×26	(21,55)	(51,34)	(55,76)
52×52	(13,21)	(13,56)	(17,43)

Model quantization methods can be divided into quantization-aware training and post-training quantization, where post-training quantization methods are divided into hybrid quantization, 8-bit integer quantization, and half-precision floating-point quantization. This paper uses the TensorRT acceleration engine to process the model weight file using the post-training quantization method, converts the weight from float type to int8 type, and performs overall optimization through a series of operations such as tensor fusion, kernel adjustment, and multi-stream execution. The algorithms can be deployed directly on embedded devices.

Results and discussion

Training dataset

This paper is a collection of flame and smoke images including single flames and smoke, multiple flames and smoke, indoor fires, forest fires, and complex background fires. Smoke is individually labeled. A total of

Fig. 4 Part of the dataset

Table 4 Software and hardware configuration

Component	Configuration
Operating system	Ubuntu 18.04
Memory	32
GPU	Nvidia GeForce RTX 3070
GPU acceleration library	CUDA 11.2 cuDNN v8.2.1
Deep learning framework	Tensorflow2.5
Programming language	Python3.9

29,980 datasets were collected and divided into training, validation, and test sets in a 7:1:2 ratio, as shown in Table 2. Datasets were selected and merged from several publicly available datasets, including FLAME (Shamsoshoara et al. 2021), FireNet Dataset (Jadon et al. 2019), and BoWFire. If the experiments below do not specify what dataset is used, then the dataset used in the experiments is the test set. Figure 4 shows some of the datasets used in this paper.

Anchor box

The prior box in the MobileNetV3-large-YOLOV4 algorithm requires two categories of flame and smoke, which are obtained by the *K*-means clustering method in this paper. The size of the input image is 416×416. When *K*-means clustering is used for 76 and 73 iterations, the ratio of the prior frame to the real frame of the flame and smoke reaches 76.54% and 74.6%, respectively. The resulting flame and the smoke prior box are shown in Table 3.

Model building and training

The specific hardware and software configuration is shown in Table 4. The network model training is based on the deep learning framework of Tensorflow 2.5, and the algorithm in this paper is implemented.

Evaluation criteria

The test set is divided into two categories, positive samples and negative samples. TP is the number of positive samples predicted as positive; FP is the number of negative samples predicted as positive; FN is the number of positive samples predicted as negative; TN is the number of negative samples predicted as negative. This paper uses the accuracy, detection rate, false detection rate, precision mAP, and running frame rate FPS as the evaluation indicators of the algorithm. The above indicators are defined as follows:

(1) Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{13}$$

(2) Detection rate (recall rate)

$$\text{Recall} = \frac{TP}{TP + FN} \tag{14}$$

(3) Missing detection rate

$$FN_{\text{rate}} = \frac{FN}{FN + TP} \tag{15}$$

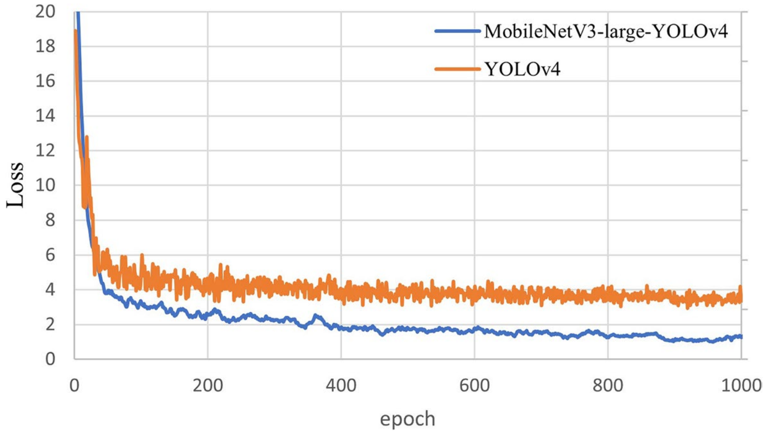


Fig. 5 The loss curves

Table 5 Ablation experiment results

Number	MobileNetV3-large	Vision transformer	Path fusion improvement for PANet	ECA Net	Add more SPP	Give PANet expansion from three to four layers	mAP(%)	FPS
1 (YOLOv4)	–	–	–	–	–	–	88.57	48
2	+	–	–	–	–	–	83.78	90
3	–	+	–	–	–	–	90.73	46
4	–	–	+	–	–	–	89.51	47
5	–	–	–	+	–	–	89.48	47
6	–	–	–	–	+	–	89.61	47
7	–	–	–	–	–	+	90.31	43
8	+	+	–	–	–	–	85.64	84
9	+	–	+	–	–	–	84.63	87
10	+	–	–	+	–	–	86.63	88
11	+	–	–	–	+	–	84.46	88
12	+	–	–	–	–	+	85.51	78
13	–	+	+	–	–	–	91.43	41
14	–	+	–	+	–	–	92.47	45
15	–	+	–	–	+	–	91.33	46
16	–	+	–	–	–	+	92.24	40
17	–	–	+	+	–	–	91.55	44
18	–	–	+	–	+	–	90.67	45
19	–	–	+	–	–	+	92.43	39
20	–	–	–	+	+	–	92.15	45
21	–	–	–	+	–	+	94.32	40
22	+	+	+	–	–	–	86.67	77
23	+	+	–	+	–	–	87.12	74
24	+	+	–	–	+	–	86.34	77
25	+	+	–	–	–	+	87.12	75
26	+	–	+	+	–	–	86.31	80
27	+	–	+	–	+	–	85.41	83
28	+	–	+	–	–	+	87.43	76
29	+	–	–	+	+	–	86.91	86
30	+	–	–	+	–	+	87.12	77
31	+	–	–	–	+	+	87.61	73
32	–	+	+	+		–	92.13	37
34	–	+	+	–	+	–	92.34	39
35	–	+	+	–	–	+	94.10	35
36	–	+	–	+	+	–	93.23	43
37	–	+	–	+	–	+	95.33	38
38	–	+	–	–	+	+	94.13	40
39	–	–	+	+	+	–	92.41	42
40	–	–	+	+	–	+	93.54	38
41	–	–	+	–	+	+	93.67	39
42	–	–	–	+	+	+	95.13	39
43	+	+	+	+	–	–	88.12	72
44	+	+	+	–	+	–	87.45	73
45	+	+	+	–	–	+	89.11	70
46	+	+	–	+	+	–	87.11	72
47	+	+	–	+	–	+	90.03	65
48	+	+	–	–	+	+	89.13	67
49	+	–	+	+	+	–	87.13	75

Table 5 (continued)

Number	MobileNetV3-large	Vision transformer	Path fusion improvement for PANet	ECA Net	Add more SPP	Give PANet expansion from three to four layers	mAP(%)	FPS
50	+	–	+	+	–	+	89.11	73
51	+	–	+	–	+	+	88.17	78
52	+	–	–	+	+	+	89.88	80
53	–	+	+	+	+	–	93.02	35
54	–	+	+	+	–	+	95.02	30
55	–	+	+	–	+	+	95.46	35
56	–	+	–	+	+	+	96.02	37
57	–	–	+	+	+	+	95.40	36
58	+	+	+	+	+	–	89.03	68
59	+	+	+	+	–	+	92.01	64
60	+	+	+	–	+	+	90.32	68
61	+	+	–	+	+	+	90.23	66
62	+	–	+	+	+	+	90.34	69
63	–	+	+	+	+	+	95.55	30
64 (our algorithm)	+	+	+	+	+	+	90.30	61

(4) False detection rate

$$\mathrm{FP_{rate}}=\frac{\mathrm{FP}}{\mathrm{FP+TN}} \tag{16}$$

(5) Precision mAP

$$\mathrm{mAP}=\frac{\sum \mathrm{AP}}{N(\mathrm{class})}=\frac{\sum \mathrm{AP}}{2} \tag{17}$$

The definition of mAP is shown in Eq. 17, which represents the average precision of the target average precision AP (AP is calculated by the P-R curve) of N classes, and $N=2$ in this experiment.

(6) Running frame rate FPS refers to the number of frames per second.

Experimental results and analysis

Speeds up the convergence of the network during training

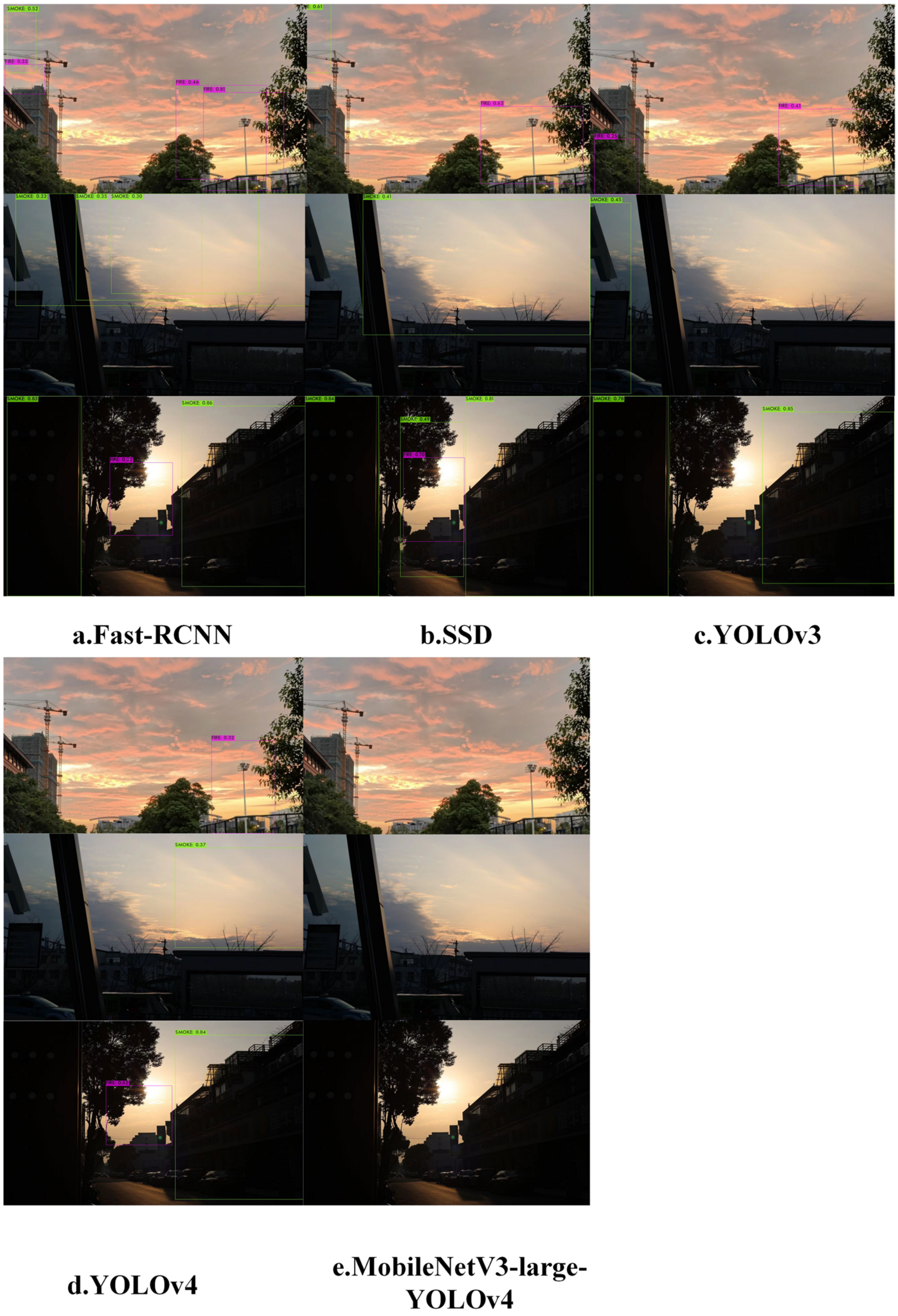
In the "Anchor box" section, the algorithm uses the *K*-means clustering algorithm to regenerate the prior box of the network, *x*. The specific loss curved of this paper's algorithm compared with YOLOv4 during the training process is shown in Fig. 5. When the training reaches 600 rounds, the algorithm in this paper has basically reached stability, while the YOLOv4 algorithm has been in a slightly oscillating state. This proves to a certain extent that the convergence speed of this paper's algorithm is significantly higher than that of YOLOv4, and the final loss value is also much lower than that of YOLOv4.

Ablation experiment

The ablation experiments were performed for the MobileNetV3-large-YOLOv4 algorithm. The images used in this experiment came from a collection of 2000 randomly selected images containing fire and smoke from the test set. As can be seen from Table 5, using MobileNetV3-large instead of CSPDarknet results in a slight decrease in mAP but a significant increase in FPS, e.g., experiment 2. Adding a path from the backbone

Table 6 Comparison of the recognition effects of various model files for similar fire and smoke

Method	False alarm rate (%)	Accuracy (%)
Types of recognition	Fire	
Faster R-CNN (Ren et al. 2015)	33.1%	67.1%
SSD (Liu et al. 2016)	27.3%	68.9%
YOLOv3 (Redmon et al. 2018)	31.2%	68.9%
YOLOv4	13.1%	74.3%
MobileNetV3-large-YOLOv4	11.1%	85.8%
Types of recognition	Smoke	
Faster R-CNN	35.6%	63.7%
SSD	33.6%	62.9%
YOLOv3	35.3%	64.7%
YOLOv4	16.7%	74.5%
MobileNetV3-large-YOLOv4	15.8%	76.3%



a.Fast-RCNN

b.SSD

c.YOLOv3

d.YOLOv4

e.MobileNetV3-large-YOLOv4

Fig. 6 Recognition renderings

to the PANet results in a slight decrease in FPS but an increase in mAP, e.g., experiment 7. Modifying the network structure of the PANet along the lines of BiFPN results in a large increase in mAP but a slight decrease in FPS, e.g., experiment 4. The introduction of more SPPs has a greater impact on mAP and a small reduction in operating speed, e.g., experiment 6. MobileNetV3-large, the BiFPN-based PANet, ECANet, and the SPP module introduced at multiple ends each have their own focus on algorithm improvement and complement each other. As a result, the MobileNetV3-large-YOLOv4 algorithm proposed in this paper achieves good overall performance. For example, in experiment 64, the algorithm achieves an mAP of 90.30% and an FPS of 61 and can accurately identify smoke and flames in real time.

Detection performance of fire-like and smoke-like targets

Due to the specific nature of detection targets such as flame and smoke, flame-like lighting effects and white cloud-like smoke effects are often encountered in real

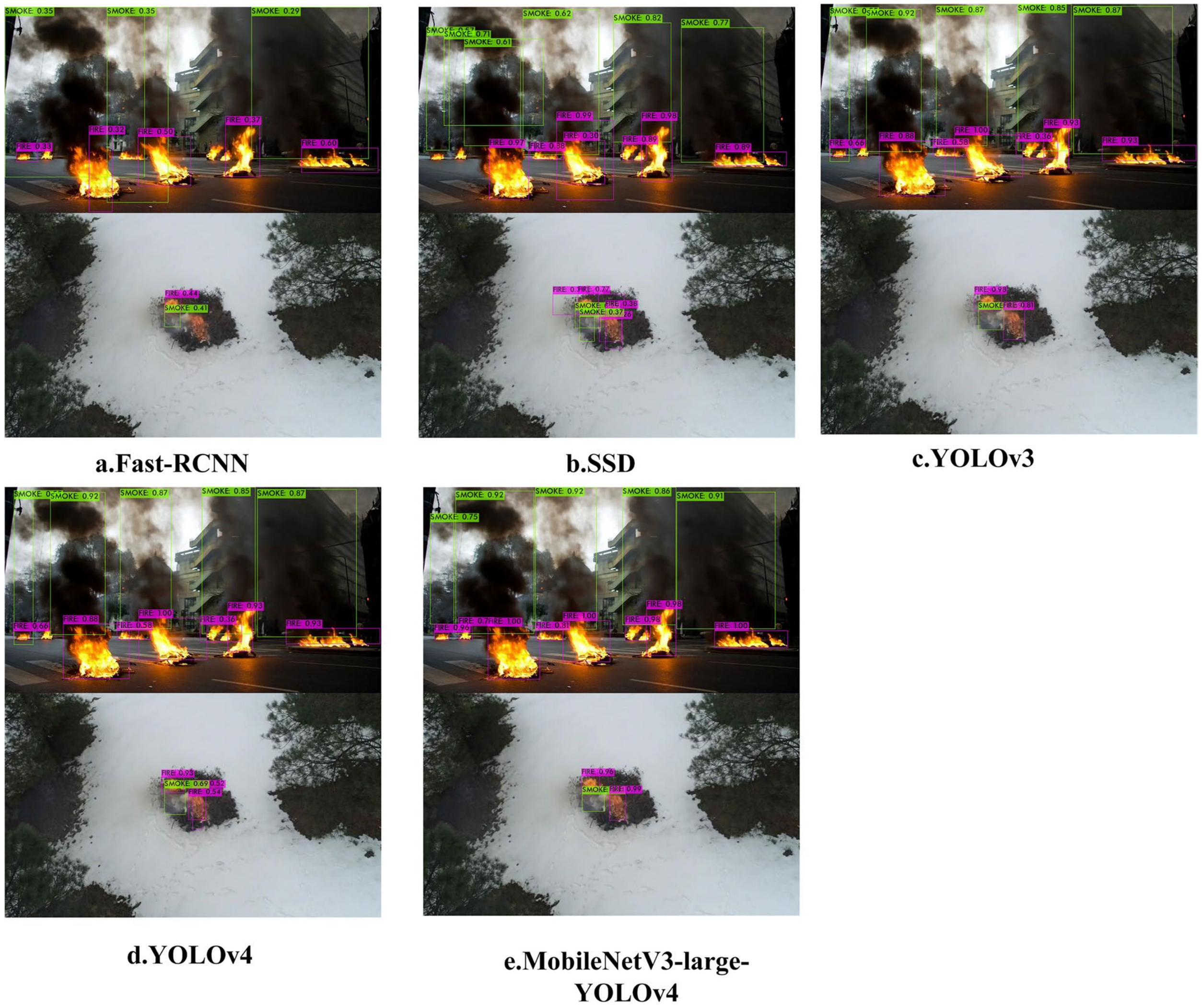
Table 7 Comparison of different methods

Method	Fire		Smoke	
	Missing detection rate (%)	Accuracy (%)	Missing detection rate (%)	Accuracy (%)
Faster R-CNN	42.3%	68.1%	43.1%	64.5%
SSD	41.7%	67.7%	43.4%	62.1%
YOLOv3	36.1%	69.9%	40.3%	65.3%
YOLOv4	24.1%	74.9%	27.3%	70.4%
MobileNetV3-large-YOLOv4	23.2%	77.7%	21.4%	74.5%

Table 8 Comparison of different methods

Method	Params (M)	mAP (%)	mAP@0.5 (%)	FPS
Faster R-CNN	108	75.56	66.03	16
SSD	90.57	76.41	69.17	60
YOLOv3	234.67	84.12	69.13	51
YOLOv4-tiny	22.57	75.13	60.13	151
YOLOv4	243.91	88.15	70.61	48
MobileNetV3-large-YOLOv4	43.71	89.73	70.33	81

fire detection scenarios. The presence of these smoke and fire targets can affect the accuracy of model detection. Considering the improvements to the YOLOv4 algorithm in this paper, this problem can be addressed to a large extent by comparing the detection effectiveness of each model file through experiments on a certain number of collected fire-like and smoke-like datasets, as shown in Table 6. It is also clear from the data in this table that the algorithm in this paper has a much lower false alarm rate for flame and smoke than


a.Fast-RCNN

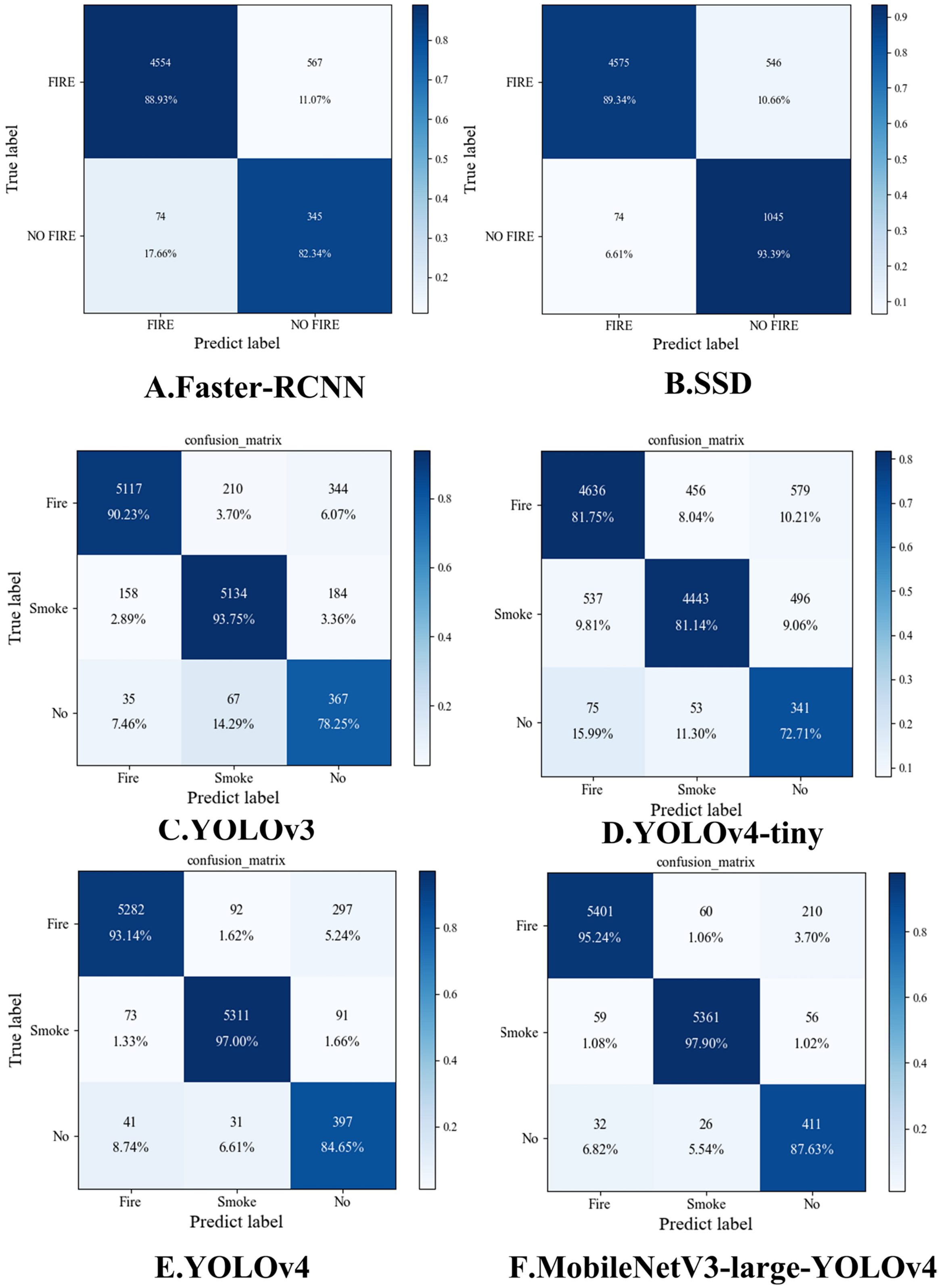
b.SSD

c.YOLOv3

d.YOLOv4

e.MobileNetV3-large-YOLOv4

Fig. 7 Recognition renderings



A.Faster-RCNN

B.SSD

C.YOLOv3

D.YOLOv4-tiny

E.YOLOv4

F.MobileNetV3-large-YOLOv4

Fig. 8 Confusion matrix for six algorithms

Table 9 Comparison of different methods

Method	Detection rate (%)	False alarm rate (%)	Accuracy (%)
Muhammad et al. (2018b)	97.48	18.69	89.82
Muhammad et al. (2018c)	93.28	9.34	92.04
Chaoxia et al. (2020)	92.44	5.61	93.36
MobileNetV3-large-YOLOv4	94.13	6.17	95.14

the other four algorithms and a much higher accuracy rate than the other algorithms. Figure 6 shows the results of the algorithm runs, from which we can see that only the algorithm in this paper did not identify fire and smoke-like scenes as flames and smoke, effectively avoiding the interference of the environment to the algorithm in this paper. From this, we deduce that the original Vision Transformer and ECA Net do have a very strong ability to filter out interference information.

The algorithm in this paper improves the detection effect of small targets

In this paper, we connect PANet structures on four effective layers and use multiple SPP structures as transition layers between the convolutional and fully connected layers to address the size mismatch and improve the algorithm's detection of small flame states early in the fire. As a result, a certain number of small target fire tests were collected. This image set was used to compare the recognition effectiveness of the model algorithm with that of detecting small-size flames or smoke. As shown in Table 7 and Fig. 7, the accuracy of the algorithm in this paper is far superior to other algorithms. We can also see from Fig. 7 that all the algorithms except this one miss some small size flames and smoke. These combined experimental analyses demonstrate that the improvements to the algorithm's neck network in this paper can indeed greatly improve the detection of small targets.

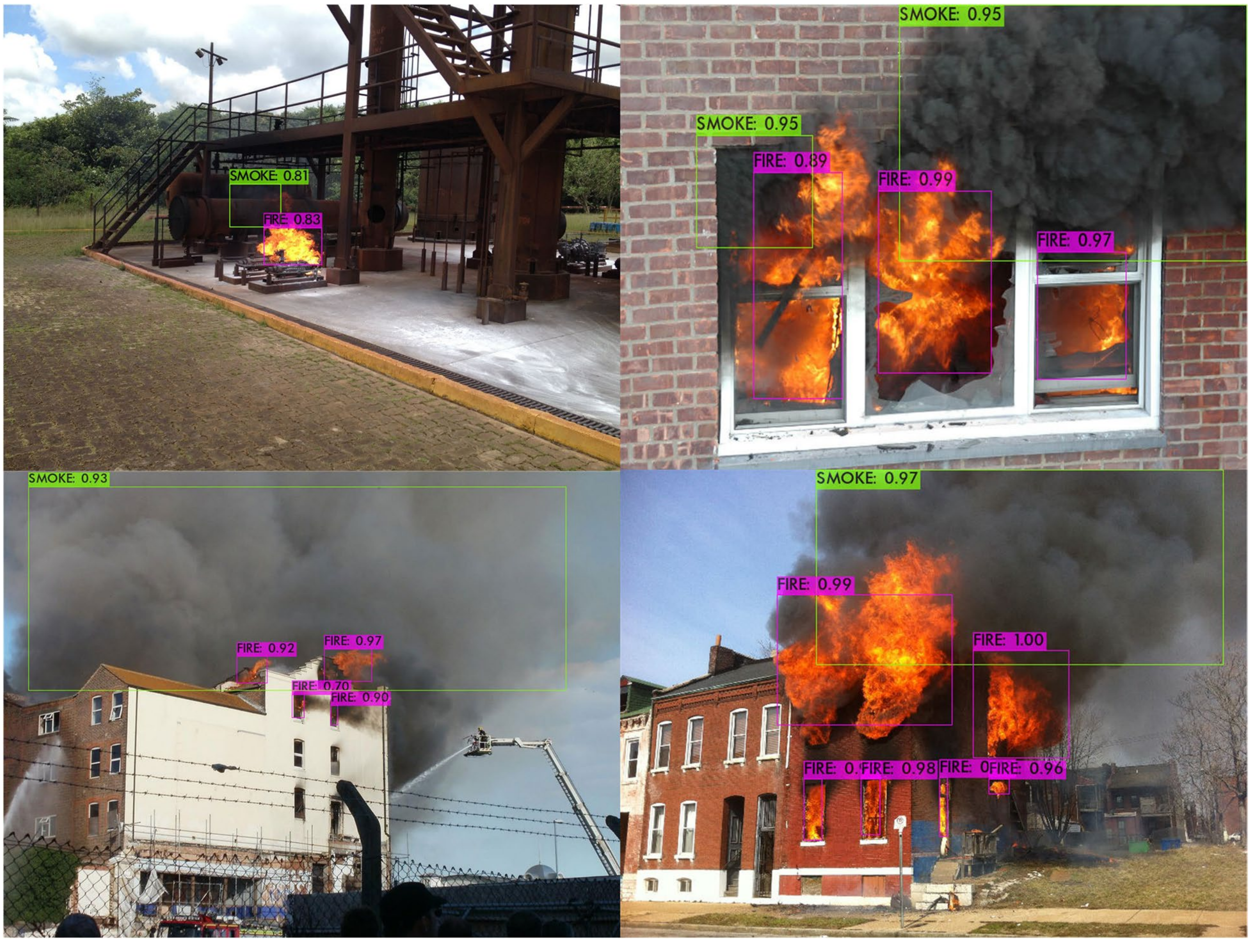


Fig. 9 Running renderings on PC

Comparison with other algorithms

In this paper, six common deep learning image recognition algorithms are used for fire detection, and the final comparison results are shown in Table 8 below. The results show that the algorithm in this paper can achieve the best balance between recognition speed and accuracy. It is only slower than YOLOv4-tiny, while its accuracy is infinitely close to YOLOv4. Considering that the difference in algorithm effectiveness cannot be visually compared by just a few percentages of data, this paper uses confusion matrices (Fig. 8) for further comparison. From the distribution of each confusion matrix, we can clearly see that the confusion matrix of this paper's algorithm has the best data on the positive diagonal, further showing the advantage of this paper's algorithm.

At the same time, this paper selects three classic algorithms (listed in Table 9) and compares them with the MobileNetV3-large-YOLOv4 algorithm. The public dataset used is BoWFire (including 119 fire images and 107 non-fire images), which has been used as a test dataset by many fire detection research works (Howard et al. 2017). We can see that the false alarm rate of MobileNetV3-large-YOLOv4 is slightly higher, but the detection rate and accuracy are better. Figure 9 shows the final recognition effect.

Algorithms are deployed on Jetson NX

We deployed the algorithms in this paper on the fire extinguishing robot RXR-M80D-13KT, as shown in Fig. 10. We play a video from a mobile device to simulate a real-time fire situation, while using the TensorRT algorithm on the embedded device (Jeston Xavier NX) to accelerate the algorithm in this paper to recognize the fire images captured from the fire extinguishing robot. We can finally find that the algorithm has achieved a frame rate of 26.13 FPS for real-time recognition and detection, and there are no significant false detections. We used a firefighting robot as the image delivery platform and Jeston Xavier NX as the algorithm running platform, and then recognized in real time a total of 2334 images, including 1387 flame images, 1082 smoke images and 846 images without flame and smoke, selected from the image test set and some real collected images, and presented the final test results on the confusion matrix (Fig. 11), and the final results are excellent. Some of the recognition results are shown in Fig. 12, and from this figure, we can also find that the algorithm did not show any misjudgment or omission.

Conclusions

This paper presents the MobileNetV3-large-YOLOv4 algorithm, which can be used for real-time fire identification in small embedded devices. Based on the experimental results, the following conclusions can be drawn.

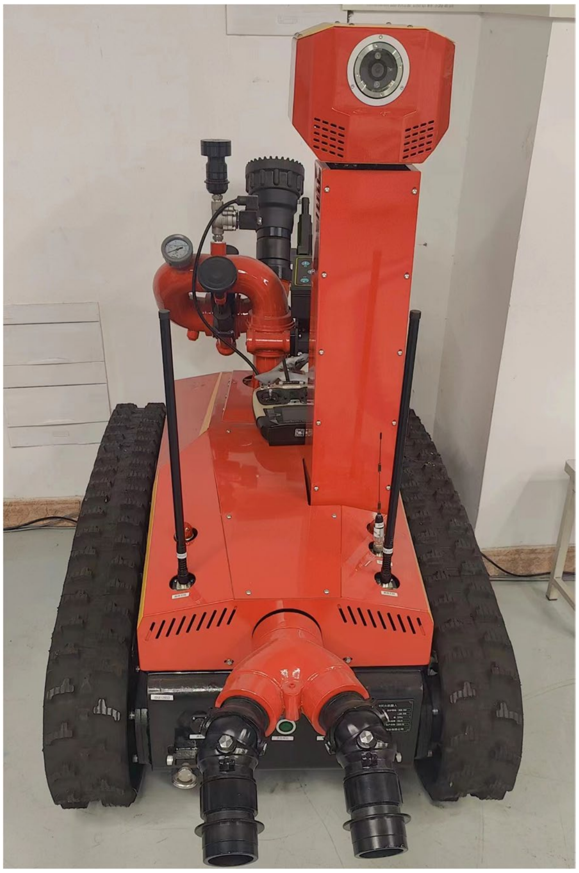


Fig. 10 Appearance of RXR-M80D-13KT

(1) Using the MobileNetV3-large-YOLOv4 algorithm to identify the fire public dataset BoWFire, the identification accuracy can reach 96.24%. Deployed on the Jeston Xavier NX, the FPS can be stabilized at around 26. Overall, this algorithm achieves a balance of running speed and accuracy, with excellent overall performance.

(2) The MobileNetV3-large-YOLOv4 algorithm has good fire recognition performance and can recognize various types of fires. Several improved and important components of the algorithm play an important role in real-time fire recognition, and through the effective integration of these components, the algorithm shows high accuracy and real-time performance.

(3) The MobileNetV3-large-YOLOv4 algorithm is not only suitable for the PC side but also for the embedded side. The algorithm can be deployed directly on the embedded Jeston Xavier NX plat-

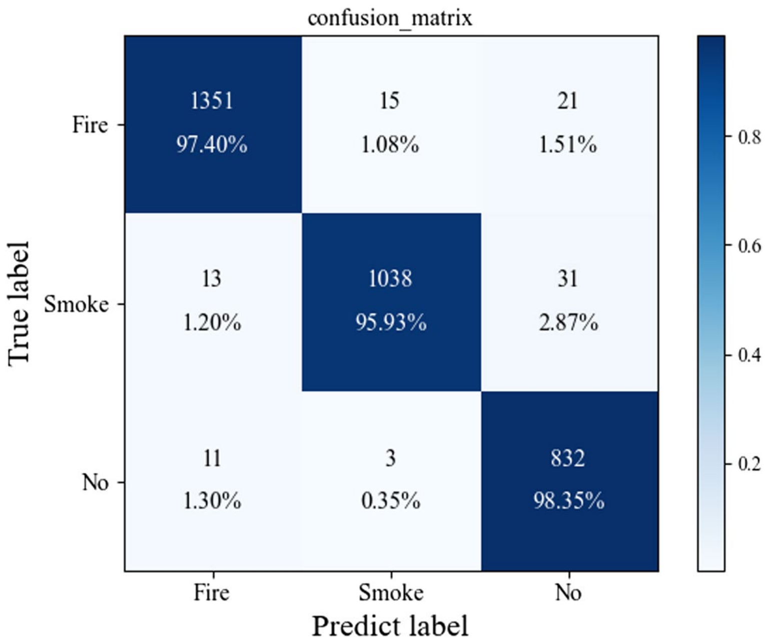


Fig. 11 The final test results on the confusion matrix

form and can meet the real time and accuracy of fire recognition. The cloud AI algorithm can therefore be pushed to the edge side for computation, which is in line with current requirements for edge intelligence.

(4) However, the algorithm is still inadequate: due to legal provisions such as fire prevention in urban areas, the algorithm in this paper lacks more field exercises, and it is hoped that more sites will be available for simulating realistic fires in the future.

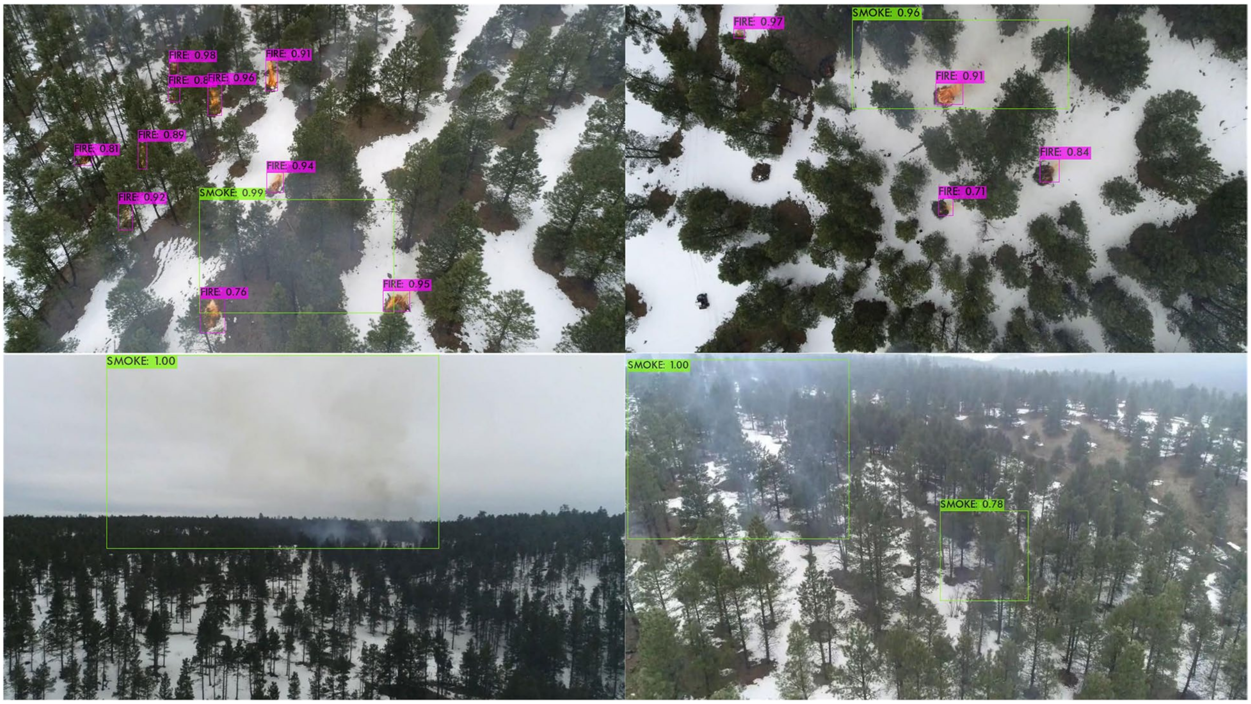


Fig. 12 The real-time recognition effect of fire

Authors' contributions

H.Z. and Y.L. conceived the idea. H.Z., J.D., Y.D., and Y.L. designed the research methods. H.Z., J.D., and Y.D. coordinated the data collection and assembly. H.Z., J.D., and Y.D. wrote the manuscript. All authors contributed to the editing and revision of the manuscript. The authors read and approved the final manuscript.

Funding

This work received support from the Industry University Research Innovation Fund of Science And Technology Development Center of the Ministry of Education (NO. 2021JQR004), Public Welfare Projects in Zhejiang Province (No. LGF20F030002), Project of Hangzhou Science and Technology Bureau (No. 20201203B96), the Ministry of Education Industry-University Cooperation Collaborative Education Project (202102019039), and Zhejiang University City College Scientific Research Cultivation Fund Project (J-202223).

Availability of data and materials

There were no known competing financial interests or personal relationships that may have affected this work.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 8 December 2022 Accepted: 12 April 2023
Published online: 15 May 2023

References

Bhattarai, M., and M. Martinez-Ramon. 2020. A deep learning framework for detection of targets in thermal images to improve firefighting. *IEEE Access* 8: 88308–88321. https://doi.org/10.1109/ACCESS.2020.2993767.

Bochkovskiy, A., C.-Y. Wang and H.-Y. M. Liao. 2020. Yolov4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 . https://doi.org/10.48550/arXiv.2004.10934.

Celik, T., and H. Demirel. 2009. Fire detection in video sequences using a generic color model. *Fire Safety Journal* 44 (2): 147–158. https://doi.org/10.1016/j.firesaf.2008.05.005.

Chaoxia, C., W. Shang, and F. Zhang. 2020. Information-guided flame detection based on faster R-CNN. *IEEE Access* 8: 58923–58932. https://doi.org/10.1109/ACCESS.2020.2982994.

Chen, T.-H., Y.-H. Yin, S.-F. Huang and Y.-T. Ye. 2006. The smoke detection for early fire-alarming system base on video processing. 2006 International Conference on Intelligent Information Hiding and Multimedia, IEEE. https://doi.org/10.1109/IIH-MSP.2006.265033

Chino, D. Y., L. P. Avalhais, J. F. Rodrigues and A. J. Traina. 2015. Bowfire: detection of fire in still images by integrating pixel color and texture analysis. 2015 28th SIBGRAPI conference on graphics, patterns and images, IEEE. https://doi.org/10.1109/SIBGRAPI.2015.19

Chunyu, Y., Z. Yongming, F. Jun and W. Jinjun (2009). Texture analysis of smoke for real-time fire detection. 2009 second international workshop on computer science and engineering, IEEE. https://doi.org/10.1109/WCSE.2009.864

Dimitropoulos, K., P. Barmpoutis, and N. Grammalidis. 2014. Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection. *IEEE Transactions on Circuits and Systems for Video Technology* 25 (2): 339–351. https://doi.org/10.1109/TCSVT.2014.2339592.

Dimitropoulos, K., P. Barmpoutis, and N. Grammalidis. 2016. Higher order linear dynamical systems for smoke detection in video surveillance applications. *IEEE Transactions on Circuits and Systems for Video Technology* 27 (5): 1143–1154. https://doi.org/10.1109/TCSVT.2016.2527340.

Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold and S. Gelly. 2020. An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 . https://doi.org/10.48550/arXiv.2010.11929

Emmy Prema, C., S. Vinsley, and S. Suresh. 2018. Efficient flame detection based on static and dynamic texture analysis in forest fire detection. *Fire Technology* 54: 255–288. https://doi.org/10.1007/s10694-017-0683-x.

Frizzi, S., R. Kaabi, M. Bouchouicha, J.-M. Ginoux, E. Moreau and F. Fnaiech. 2016. Convolutional neural network for video fire and smoke detection. IECON 2016–42nd Annual Conference of the IEEE Industrial Electronics Society, IEEE. https://doi.org/10.1109/IECON.2016.7793196

Genovese, A., R. D. Labati, V. Piuri and F. Scotti (2011). Wildfire smoke detection using computational intelligence techniques. 2011 IEEE international conference on computational intelligence for measurement systems and applications (CIMSA) proceedings, IEEE. https://doi.org/10.1109/CIMSA.2011.6059930

Gong, D., T. Ma, J. Evans, and S. He. 2021. Deep neural networks for image super-resolution in optical microscopy by using modified hybrid task cascade U-Net. *Progress in Electromagnetics Research* 171: 185–199. https://doi.org/10.2528/PIER21110904.

Gunay, O., B.U. Toreyin, K. Kose, and A.E. Cetin. 2012. Entropy-functional-based online adaptive decision fusion framework with application to wildfire detection in video. *IEEE Transactions on Image Processing* 21 (5): 2853–2865. https://doi.org/10.1109/TIP.2012.2183141.

Günay, O. and A. E. Çetin. 2015. Real-time dynamic texture recognition using random sampling and dimension reduction. 2015 IEEE International Conference on Image Processing (ICIP), IEEE. https://doi.org/10.1109/ICIP.2015.7351371

Habiboğlu, Y.H., O. Günay, and A.E. Çetin. 2012. Covariance matrix-based fire and flame detection method in video. *Machine Vision and Applications* 23: 1103–1113. https://doi.org/10.1007/s00138-011-0369-1.

Han, D., and B. Lee. 2009. Flame and smoke detection method for early real-time detection of a tunnel fire. *Fire Safety Journal* 44 (7): 951–961. https://doi.org/10.1016/j.firesaf.2009.05.007.

He, K., X. Zhang, S. Ren, and J. Sun. 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37 (9): 1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824.

Hongyu, H., K. Ping, F. Li and S. Huaxin. 2020. An improved multi-scale fire detection method based on convolutional neural network. 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), IEEE. https://doi.org/10.1109/ICCWAMTIP51612.2020.9317360

Howard, A. G., M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam. 2017. Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 . https://doi.org/10.48550/arXiv.1704.04861

Howard, A., M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang and V. Vasudevan (2019). Searching for mobilenetv3. Proceedings of the IEEE/CVF international conference on computer visionhttps://doi.org/10.1109/ICCV.2019.00140

Hu, J., L. Shen and G. Sun. 2018. Squeeze-and-excitation networks. Proceedings of the IEEE conference on computer vision and pattern recognition. https://doi.org/10.1109/TPAMI.2019.2913372

Huang, Z., J. Wang, X. Fu, T. Yu, Y. Guo, and R. Wang. 2020. DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection. *Information Sciences* 522: 241–258. https://doi.org/10.1016/j.ins.2020.02.067.

Huang, J., Z. He, Y. Guan, and H. Zhang. 2023. Real-time forest fire detection by ensemble lightweight YOLOX-L and defogging method. *Sensors* 23 (4): 1894. https://doi.org/10.3390/s23041894.

Jadon, A., M. Omama, A. Varshney, M. S. Ansari and R. Sharma. 2019. FireNet: a specialized lightweight fire & smoke detection model for real-time IoT applications. arXiv preprint arXiv:1905.11922 . https://doi.org/10.48550/arXiv.1905.11922

Jia, Y., J. Yuan, J. Wang, J. Fang, Q. Zhang, and Y. Zhang. 2016. A saliency-based method for early smoke detection in video sequences. *Fire Technology* 52: 1271–1292. https://doi.org/10.1007/s10694-014-0453-y.

Kim, J.-H., and B.Y. Lattimer. 2015. Real-time probabilistic classification of fire and smoke using thermal imagery for intelligent firefighting robot. *Fire Safety Journal* 72: 40–49. https://doi.org/10.1016/j.firesaf.2015.02.007.

Kim, B., and J. Lee. 2019. A video-based fire detection using deep learning models. *Applied Sciences* 9 (14): 2862. https://doi.org/10.3390/app9142862.

Kim, Y.-H., A. Kim, and H.-Y. Jeong. 2014. RGB color model based the fire detection algorithm in video sequences on wireless sensor network. *International Journal of Distributed Sensor Networks* 10 (4): 923609. https://doi.org/10.1155/2014/923609.

Kim, J.-H., S. Jo and B. Y. Lattimer. 2016. Feature selection for intelligent firefighting robot classification of fire, smoke, and thermal reflections using thermal infrared images. Journal of Sensors 2016. https://doi.org/10.1155/2016/8410731

Lin, G., Y. Zhang, G. Xu, and Q. Zhang. 2019. Smoke detection on video sequences using 3D convolutional neural networks. *Fire Technology* 55: 1827–1847. https://doi.org/10.1007/s10694-019-00832-w.

Lin, T.-Y., P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie. 2017. Feature pyramid networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition. https://doi.org/10.48550/arXiv.1612.03144

Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg. 2016. SSD: single shot multibox detector. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, Springer. https://doi.org/10.48550/arXiv.1512.02325

Liu, S., L. Qi, H. Qin, J. Shi and J. Jia. 2018. Path aggregation network for instance segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition. https://doi.org/10.48550/arXiv.1803.01534

Mao, Q.-C., H.-M. Sun, L.-Q. Zuo, and R.-S. Jia. 2020. Finding every car: A traffic surveillance multi-scale vehicle object detection method. *Applied Intelligence* 50: 3125–3136. https://doi.org/10.1007/s10489-020-01704-5.

Muhammad, K., J. Ahmad, and S.W. Baik. 2018a. Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing* 288: 30–42. https://doi.org/10.1016/j.neucom.2017.04.083.

Muhammad, K., J. Ahmad, I. Mehmood, S. Rho, and S.W. Baik. 2018b. Convolutional neural networks based fire detection in surveillance videos. *IEEE Access* 6: 18174–18183. https://doi.org/10.1109/ACCESS.2018.2812835.

Muhammad, K., J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S.W. Baik. 2018c. Efficient deep CNN-based fire detection and localization in video surveillance applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49 (7): 1419–1434. https://doi.org/10.1109/TSMC.2018.2830099.

Redmon, J. and A. Farhadi. 2018. Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767 . https://doi.org/10.48550/arXiv.1804.02767

Ren, S., K. He, R. Girshick and J. Sun. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. Advances in neural information processing systems 28. https://doi.ieeecomputersociety.org/10.1109/TPAMI.2016.2577031

Sandler, M., A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen. 2018. Mobilenetv2: inverted residuals and linear bottlenecks. Proceedings of the IEEE conference on computer vision and pattern recognition. https://doi.org/10.48550/arXiv.1801.04381

Sathishkumar, V.E., J. Cho, M. Subramanian, and O.S. Naren. 2023. Forest fire and smoke detection using deep learning-based learning without forgetting. *Fire Ecology* 19 (1): 1–17. https://doi.org/10.1186/s42408-022-00165-0.

Shamsoshoara, A., F. Afghah, A. Razi, L. Zheng, P.Z. Fulé, and E. Blasch. 2021. Aerial imagery pile burn detection using deep learning: the FLAME dataset. *Computer Networks* 193: 108001. https://doi.org/10.1016/j.comnet.2021.108001.

Succetti, F., A. Rosato, F. Di Luzio, A. Ceschini and M. Panella. 2022. A fast deep Learning technique for Wi-Fi-based human activity recognition. *Progress in Electromagnetics Research* 174: 127–141. https://doi.org/10.2528/PIER22042605.

Tan, M., R. Pang and Q. V. Le. 2020. Efficientdet: scalable and efficient object detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. https://doi.org/10.48550/arXiv.1911.09070

Tao, C., J. Zhang and P. Wang. 2016. Smoke detection based on deep convolutional neural networks. 2016 International conference on industrial informatics-computing technology, intelligent technology, industrial information integration (ICIICII), IEEE. https://doi.org/10.1109/ICIICII.2016.0045

Töreyin, B.U., Y. Dedeoğlu, U. Güdükbay, and A.E. Cetin. 2006. Computer vision based method for real-time fire and flame detection. *Pattern Recognition Letters* 27 (1): 49–58. https://doi.org/10.1016/j.patrec.2005.06.015.

Töreyin, B. U., Y. Dedeoğlu and A. E. Cetin. 2005. Wavelet based real-time smoke detection in video. 2005 13th European signal processing conference, IEEE.

Wang, Q., B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu. 2020. ECA-Net: Efficient channel attention for deep convolutional neural networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR42600.2020.01155.

Wang, X., Y. Li and Z. Li. 2020 Research on flame detection algorithm based on multi-feature fusion. 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), IEEE. https://doi.org/10.1109/ITNEC48623.2020.9084825

Wu, Z., R. Xue, and H. Li. 2022. Real-time video fire detection via modified YOLOv5 network model. *Fire Technology* 58 (4): 2377–2403. https://doi.org/10.1007/s10694-022-01260-z.

Xu, G., Y. Zhang, Q. Zhang, G. Lin, Z. Wang, Y. Jia, and J. Wang. 2019. Video smoke detection based on deep saliency network. *Fire Safety Journal* 105: 277–285. https://doi.org/10.1016/j.firesaf.2019.03.004.

Xu, R., H. Lin, K. Lu, L. Cao, and Y. Liu. 2021. A forest fire detection system based on ensemble learning. *Forests* 12 (2): 217. https://doi.org/10.3390/f12020217.

Xue, Q., H. Lin, and F. Wang. 2022. FCDM: An improved forest fire classification and detection model based on YOLOv5. *Forests* 13 (12): 2129. https://doi.org/10.3390/f13122129.

Yin, Z., B. Wan, F. Yuan, X. Xia, and J. Shi. 2017. A deep normalization and convolutional neural network for image smoke detection. *IEEE Access* 5: 18429–18438. https://doi.org/10.1109/ACCESS.2017.2747399.

Yuan, F. 2008. A fast accumulative motion orientation model based on integral image for video smoke detection. *Pattern Recognition Letters* 29 (7): 925–932. https://doi.org/10.1016/j.patrec.2008.01.013.

Yuan, F., J. Shi, X. Xia, Y. Fang, Z. Fang, and T. Mei. 2016a. High-order local ternary patterns with locality preserving projection for smoke detection and image classification. *Information Sciences* 372: 225–240. https://doi.org/10.1016/j.ins.2016.08.040.

Yuan, C., Z. Liu and Y. Zhang. 2016 Vision-based forest fire detection in aerial images for firefighting using UAVs. 2016 International conference on unmanned aircraft systems (ICUAS), IEEE. https://doi.org/10.1109/ICUAS.2016.7502546

Zhang, Q.-X., G.-H. Lin, Y.-M. Zhang, G. Xu, and J.-J. Wang. 2018. Wildland forest fire smoke detection based on faster R-CNN using synthetic smoke images. *Procedia Engineering* 211: 441–446. https://doi.org/10.1016/j.proeng.2017.12.034.

Zhao, L., L. Zhi, C. Zhao, and W. Zheng. 2022. Fire-YOLO: A small target object detection method for fire inspection. *Sustainability* 14 (9): 4930. https://doi.org/10.3390/su14094930.

Zheng, H., S. Dembélé, Y. Wu, Y. Liu, H. Chen, and Q. Zhang. 2023. A lightweight algorithm capable of accurately identifying forest fires from remotely sensed images. *Frontiers in Forests and Global Change* 6: 36. https://doi.org/10.3389/ffgc.2023.1134942.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.