frontiers | Frontiers in Forests and Global Change

TYPE Original Research
PUBLISHED 28 February 2023
DOI 10.3389/ffgc.2023.1134942

Check for updates

OPEN ACCESS

EDITED BY
Giuseppe Ruello,
University of Naples Federico II, Italy

REVIEWED BY
María Menéndez-Miguélez,
Instituto Nacional de Investigación y Tecnología Agroalimentaria (INIA), Spain
Donato Amitrano,
Italian Aerospace Research Centre, Italy
Francescopaolo Sica,
Munich University of the Federal Armed Forces, Germany

*CORRESPONDENCE
Yan Liu
✉ liuy@zucc.edu.cn

SPECIALTY SECTION
This article was submitted to Fire and Forests, a section of the journal Frontiers in Forests and Global Change

RECEIVED 31 December 2022
ACCEPTED 13 February 2023
PUBLISHED 28 February 2023

CITATION
Zheng H, Dembélé S, Wu Y, Liu Y, Chen H and Zhang Q (2023) A lightweight algorithm capable of accurately identifying forest fires from UAV remote sensing imagery. *Front. For. Glob. Change* 6:1134942. doi: 10.3389/ffgc.2023.1134942

A lightweight algorithm capable of accurately identifying forest fires from UAV remote sensing imagery

Hongtao Zheng[1], Sounkalo Dembélé[2], Yongxin Wu[2], Yan Liu[1]*, Hongli Chen[1,3] and Qiujie Zhang[1,4]

[1]School of Information and Electrical Engineering, Hangzhou City University, Hangzhou, China, [2]FEMTO-ST, University Bourgogne Franche-Comté, CNRS, Besançon, France, [3]School of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China, [4]Zhejiang Dahua Technology Co., Ltd., Hangzhou, China

Forest fires often have a devastating effect on the planet's ecology. Accurate and rapid monitoring of forest fires has therefore become a major focus of current research. Considering that manual monitoring is often inefficient, UAV-based remote sensing fire monitoring algorithms based on deep learning are widely studied and used. In UAV monitoring, the size of the flames is very small and potentially heavily obscured by trees, so the algorithm is limited in the amount of valid information it can extract. If we were to increase the ability of the algorithm to extract valid information simply by increasing the complexity of the algorithm, then the algorithm would run much slower, ultimately reducing the value of the algorithm to the application. To achieve a breakthrough in both algorithm speed and accuracy, this manuscript proposes a two-stage recognition method that combines the novel YOLO algorithm (FireYOLO) with Real-ESRGAN. Firstly, as regards the structure of the FireYOLO algorithm, "the backbone part adopts GhostNet and introduces a dynamic convolutional structure, which improves the information extraction capability of the morphologically variable flame while greatly reducing the computational effort; the neck part introduces a novel cross-layer connected, two-branch Feature Pyramid Networks (FPN) structure, which greatly improves the information extraction capability of small targets and reduces the loss in the information transmission process; the head embeds the attention-guided module (ESNet) proposed in this paper, which enhances the attention capability of small targets". Secondly, the flame region recognized by FireYOLO is input into Real-ESRGAN after a series of cropping and stitching operations to enhance the clarity, and then the enhanced image is recognized for the second time with FireYOLO, and, finally, the recognition result is overwritten back into the original image. Our experiments show that the algorithms in this paper run very well on both PC-based and embedded devices, adapting very well to situations where they are obscured by trees as well as changes in lighting. The overall recognition speed of Jeston Xavier NX is about 20.67 FPS (latency-free real-time inference), which is 21.09% higher than the AP of YOLOv5x, and are one of the best performance fire detection algorithm with excellent application prospects.

KEYWORDS

fire, FireYOLO, Real-ESRGAN, GhostNet, two-branch, FPN, ESNet

1. Introduction

Forests are potent regulators of the Earth's ecosystem (Mitchard, 2018; De Frenne et al., 2019). For example, tropical rainforests (Brancalion et al., 2019) are known as the lungs and green heart of the Earth, and are no less important to the planet than the lungs and heart are to people. Fires can easily cause irreparable damage to the environment and ecology, as exemplified by the Amazon forest fires in 2019 (Lizundia-Loiola et al., 2020) and the Australian forest fires in 2019–2020 (Ward et al., 2020), which caused very significant ecological damage. The aftermath of the fires is a reminder of ecological fragility (Barbosa et al., 2018). This has led to a growing interest in forest fire monitoring. Forest fire prediction (de Santana et al., 2021) and identification are critical research issues, and the most popular fire detection algorithms mainly include traditional manual detection, and sensor-based and machine vision-related algorithms. Among them, sensor-based detection systems (Bouabdellah et al., 2013; Sasmita et al., 2018; Sarwar et al., 2019; Cui, 2020) are more effective in smaller indoor spaces, and standard sensors include smoke sensors and temperature sensors. However, this approach has a limited detection distance, high installation costs, and complex communication and power supply networking problems.

Deep learning-based algorithms are widely used in the field of fire monitoring (Luo et al., 2018; Shen et al., 2018; Harkat et al., 2020; Li and Zhao, 2020; Wang et al., 2021; Xu et al., 2021; Zheng et al., 2022). These neural networks include some classical classification networks (Simonyan and Zisserman, 2015; Szegedy, 2015; He et al., 2016; Krizhevsky et al., 2017) and detection networks (Liu et al., 2016; Ren et al., 2017; Zhang et al., 2018). Yuan et al. (2019) constructed deep multiscale neural networks with feature extraction layers consisting of multiple parallel convolutional layers and realized multiscale feature extraction through multiscale convolutional kernels to solve the problems caused by light and scale invariance. They achieved high accuracy but using multiple convolutional blocks will inevitably increase the model complexity and make it difficult to deploy. Han et al. (2017) used background subtractive motion detection based on Gaussian mixture model combined with RGB and HSV multi-color features to detect flame elements in video sequences. Zhan et al. (2022) proposed a smoke detection algorithm based on the ARGNet structure, which combines recursive feature pyramids and ARGNet and can effectively cope with the problem of transparent smoke and inconspicuous edges. Xue et al. (2022) proposed an improved forest fire small-target detection model based on YOLOv5 with an improved backbone layer and embedded SPPFP module and added CBAM Net in YOLOv5 to improve the recognition of forest fire small-targets. These algorithms have some advantages in terms of accuracy or inference speed, but they mostly do not work perfectly on small embedded devices and these algorithms are not well-solved for problems such as illumination changes and foreign object occlusion.

Therefore, the aim of this paper is to propose a remote sensing fire detection algorithm with low computational complexity and high detection accuracy. Considering the obvious advantages of the YOLO family of algorithms (Redmon et al., 2016; Redmon and Farhadi, 2017, 2018; Bochkovskiy et al., 2020) in terms of structural plasticity and good adaptability to small embedded and other hardware devices, this paper proposes a new target detection algorithm, FireYOLO, by mimicking the YOLOv4 structure. In the backbone network, in order to improve the ability of the algorithm to detect multi-scale targets while significantly reducing the overall computational complexity of the algorithm, this paper proposes the GhostNet (Han et al., 2020) structure with embedded dynamic convolution and replaces the original CSPDarknet53 structure; in the neck structure, in order to improve the ability of the algorithm to detect multi-scale targets without increasing the complexity of the algorithm, this paper proposes a two-branch FPN (Lin et al., 2017) structure, which can increase the low sampling without increasing the depth of the structure. In the head network, we propose a two-branch parallel attention-guided module (ESNet) based on the Efficient Channel Attention [ECA (Wang et al., 2020)] and Spatial Group Enhancement [SGE (Li et al., 2019)] modules, and embed them in the head network to make the algorithm more focused on the valid information in the image. The optimization effects described above are mainly derived from assumptions made as a result of the theoretical analysis of these structures. Whether these structures lead to corresponding performance improvements in the algorithm, and whether they are compatible rather than exclusive with the underlying structure of YOLOv4, are questions that require subsequent experimental verification.

It should be noted that in UAV remote sensing images of objects from such a distance, the shape and outline of the target may be blurred, but the color is still largely discernible. This also makes the algorithm more concerned with color differences in model training and more sensitive to the color features of flames in the recognition process, which ultimately leads to a sharp increase in the false alarm rate of the algorithm for flames in remotely sensed imagery (the algorithm has a high probability of identifying targets with colors similar to flames as flames), but this situation also reduces the missing detection rate of the algorithm to some extent. Therefore, we also need to reduce the false alarm rate of the FireYOLO algorithm for remotely sensed images by using a class of algorithms that can enhance the effective information of local features. Therefore, in this paper, we choose to use the Real-ESRGAN (Wang et al., 2021) algorithm to improve the clarity of suspected fire areas in remotely sensed images. The above analytical results are still based on the structure of the algorithm and need to be demonstrated in subsequent experiments.

The details of how FireYOLO and Real-ESRGAN work together is that FireYOLO first identifies the image, then passes any areas of the image suspected of being flames to the Real-ESRGAN algorithm to improve the clarity of those areas, and finally uses FireYOLO to identify those areas a second time. However, a weakness in the operation of FireYOLO+Real-ESRGAN (the algorithm in this paper) is that if FireYOLO does not identify the flame the first time, then Real-ESRGAN will not be able to make the missed flame region clear, and then FireYOLO will not be able to identify the flame a second time. In other words, if FireYOLO cannot identify a flame in a remotely sensed image, then there is no way for the algorithm in this paper to identify it. Therefore, the miss rate of FireYOLO will directly affect the accuracy of the algorithm in this paper. However, in the previous paragraph we have analysed that the detection rate of FireYOLO will remain relatively low due to the high sensitivity of the algorithm to color. In terms of overall performance, the accuracy of the algorithm in this paper will also be very little affected by the missing detection rate as long as the leakage rate of FireYOLO is low enough. We then still need to verify the above analysis through experiments.

In summary, based on the theoretical analysis, we have every reason to believe that the algorithm in this paper is a UAV remote sensing fire detection algorithm that is highly accurate and, due to its low computational complexity, stable and fast on platforms with low computational power. In the next section we will focus on the structural principles of these algorithms.

In "section 2 Materials and methods" we will focus on the structural principles of the algorithm. In "section 3 Experimental setting" we conduct experiments to verify the validity of the theoretical analysis of the structure above and to evaluate the performance of our algorithm, and we also analyse and discuss various experimental phenomena. In "section 5 Conclusion" we summarize the conclusions drawn from the experiments.

2. Materials and methods

To further improve the real-time performance of deep learning-based forest fire detection algorithms and the detection performance of small target fires at long distances, a two-step recognition method combining FireYOLO and ESRGAN Net is proposed in this paper. First, regarding the structure of the FireYOLO algorithm, GhostNet with embedded dynamic convolution is used in the backbone part to remove redundant features of complex backgrounds, thus imparting the ability to recognize multi-featured flame patterns while greatly reducing the computational effort. A novel FPN two-branch structure is used in the neck to increase the feature pyramid level to cover the target scale, while using cross-layer connections to reduce the distance of feature transfer and reduce the loss of effective information. The head network is embedded with a novel attention-guided module (ESNet), designed in this project to enhance the ability to focus on small targets. Next, the locations of suspected small fires initially identified by FireYOLO are cropped out and fed into the enhanced super-resolution-generative adversarial Network (Real-ESRGAN) to enhance the clarity of small fires, and then the clarity-enhanced images are identified a second time with FireYOLO, and, finally, more accurate small target identification results are output. Our algorithm are able to run stably and quickly on high performance devices. Finally, to verify the suitability of the algorithm, we installed it on a Jetson Xavier NX (a small embedded device with less computing power) and the overall recognition speed was about 20.67 FPS (real-time inference), which is 21.09% higher than the AP (Average Precision) of the YOLOv5x.

To facilitate understanding of Figure 1, we will further analyze its constitutive logic. Figure 1A shows three main improvements over the original YOLOv4 algorithm, where Figure 1Aa shows the head structure of the FireYOLO algorithm, which uses the GhostNet algorithm to embed dynamic convolution, the exact structure of which is shown in Figure 1B. Figure 1Ab shows the neck structure of FireYOLO, which consists of the SPP (He et al., 2015) and the two-branch FPN structure; Figure 1Ac shows the head structure of FireYOLO embedded with the ESNet structure proposed in this paper, represented as ES in the figure, as shown in Figure 1Ca, while the Attention-guide layer structure of Figure 1Ca is shown in Figure 1Cb. Figure 1D shows the network structure of the Real-ESRGAN algorithm. These algorithm structures were run using the logic and sequence of Figure 1E.

(1) Figure 2 gives a more visual representation of the workflow of Figure 1E. The algorithm flow in this figure follows the black arrows step by step, while the blue bidirectional arrows indicate that the upper and lower diagrams are viewed in comparison. Observing these two diagrams we can see that the confidence level of the flame target in the second recognition of FireYOLO is much higher than the first recognition. The detailed flow of the entire algorithm is shown below: The first step is to capture live images of the forest remotely *via* surveillance cameras or drones;

(2) The images are scaled frame made by-frame to a size of 408 × 408, and then input into FireYOLO's GhostNet backbone feature extraction layer;

(3) The features with different semantic information contents are stitched by SPP and improved FPN to obtain four scales of 104 × 104, 52 × 52, 26 × 26, and 13 × 13;

(4) The four scales are passed through a network embedded in the ES Net, resulting in an initial recognition region. When the confidence level of the region is below a set threshold or the size of the region is small enough, the region is cropped down;

(5) The cropped multiple targets are randomly stitched together, and the photo gaps that appear after stitching are filled with white to finally form a rectangular photo;

(6) The suspected small target is input into the super-resolution algorithm (Real-ESRGAN) for feature enhancement;

(7) We then re-import the enhanced image into FireYOLO for target recognition and over-lay the final recognition result back onto the original small-area recognition result.

2.1. Introducing dynamic convolution at GhostNet

GhostNet is used as the backbone extraction network for FireYOLO, considering the dual balance of real time and accuracy, GhostNet can reasonably utilize the redundancy of feature maps and obtain better performance in terms of accuracy and latency than other lightweight networks. It can achieve excellent algorithm performance on embedded devices with ARM architecture such as Jeston Xavier NX.

GhostNet is composed of the Ghost module, which consists of ordinary convolution and cheap operations. The m original feature maps are generated by one convolution, and these original feature maps are transformed in two parts: one part uses 1×1 ordinary convolution for identity to generate m necessary feature concentrations, and the other part uses depth-separable convolution blocks for layer-by-layer convolution to linearly transform and stack the m original feature maps to generate s Ghost feature maps. The m feature maps after identity are stacked with the s Ghost feature maps to obtain n new feature maps, $n = m \times s$.

Assuming that the Ghost module contains an intrinsic feature map and $m \cdot (s-1) = \frac{n}{s} \cdot (s-1)$ linear transformation

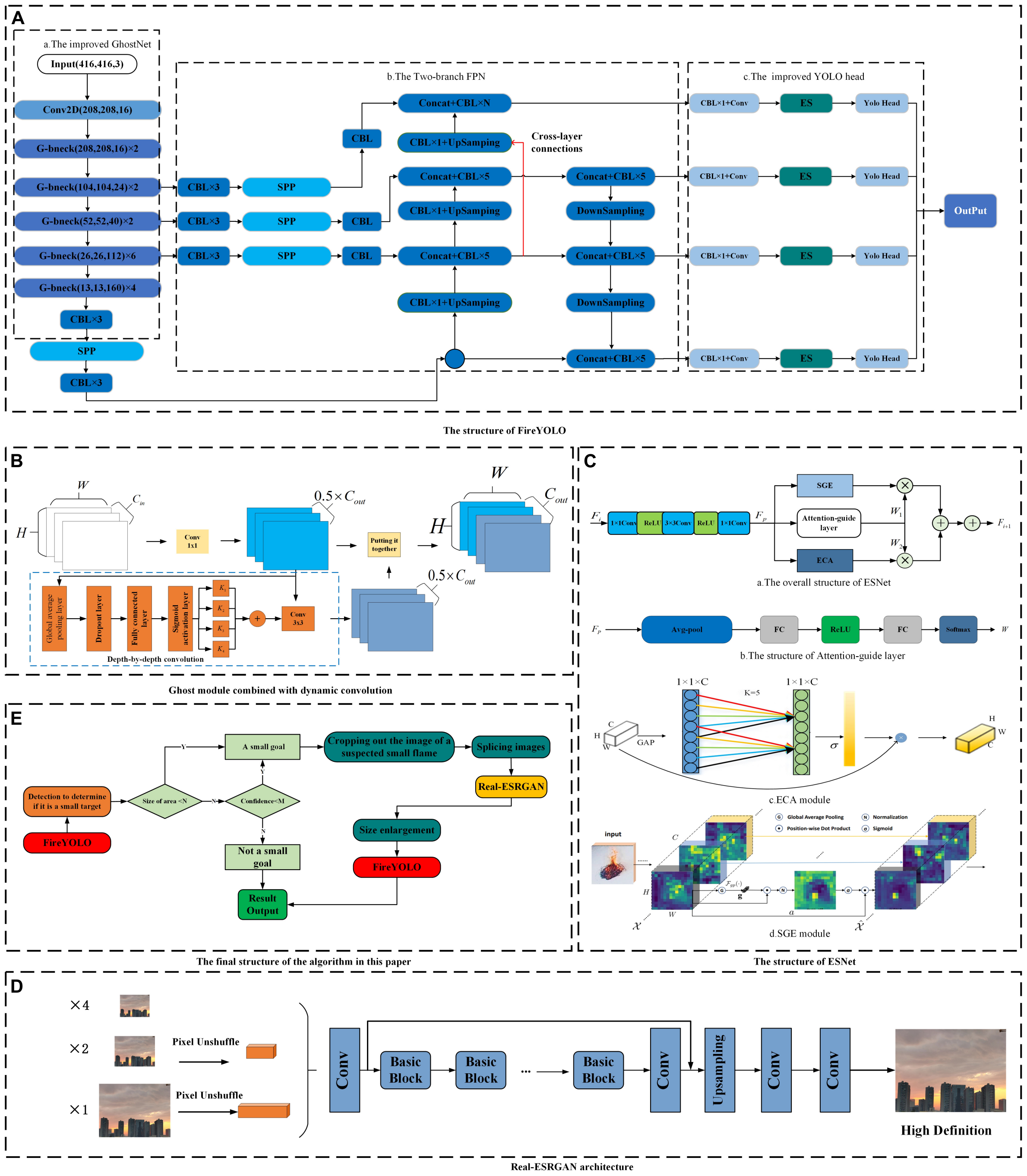


FIGURE 1
The operational logic and algorithmic structure of our algorithm. Panel **(E)** shows the overall framework of the algorithm in this paper. Panel **(A)** shows the overall structure of FireYOLO in **(E)**; **(D)** shows the structure of Real-ESRGAN in **(E)**; while **(B)** shows the structure of the Ghost module in the backbone of the FireYOLO algorithm, which embeds dynamic convolution on top of the original Ghost module; and **(C)** shows the structure of the ESNet embedded in the head network of the FireYOLO algorithm.

operations, the kernel of each operation should be $d \times d$. The theoretical speedup ratio for the Ghost module to upgrade ordinary convolution is:

$$R_S = \frac{n\cdot w'\cdot h'\cdot c\cdot k\cdot k}{\frac{n}{s}\cdot w'\cdot h'\cdot c\cdot k\cdot k+(s-1)\cdot\frac{n}{s}\cdot w'\cdot h'\cdot c\cdot d\cdot d} = \frac{c\cdot k\cdot k}{\frac{1}{s}\cdot c\cdot k\cdot k+(s-1)\cdot\frac{1}{s}\cdot d\cdot d}\approx\frac{c\cdot s}{c+s-1}\approx s \qquad (1)$$

where w' and h' are the width and height of the output image, respectively, c is the number of input channels to the convolution kernel.

Considering that the algorithm in this paper is aimed at fire detection algorithm in remote sensing images, we have improved GhostNet to some extent. Specifically, we introduce dynamic convolution to improve the Ghost module so that it can adapt to the complex and variable morphology of flames.

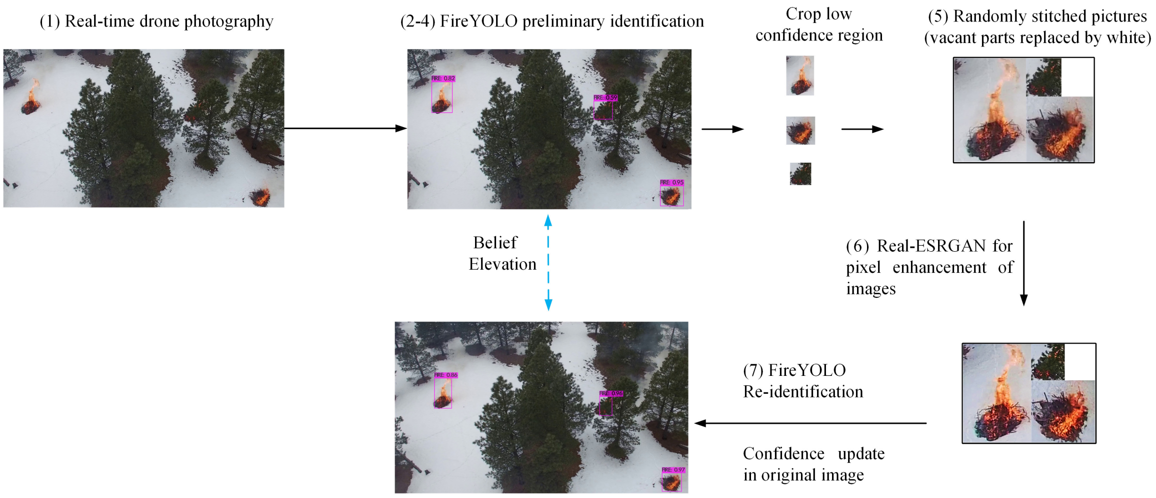

(1) Real-time drone photography
(2-4) FireYOLO preliminary identification
Crop low confidence region
(5) Randomly stitched pictures (vacant parts replaced by white)
Belief Elevation
(6) Real-ESRGAN for pixel enhancement of images
(7) FireYOLO Re-identification
Confidence update in original image

FIGURE 2
The detailed flow of the algorithm in this paper. The seven numbers in the diagram correspond to the seven stages described above.

Our inspiration for introducing dynamic convolution to make some improvements to GhostNet's Ghost module comes from the literature (Zhang et al., 2022). The dynamic convolution is calculated by weighting four convolution kernels with the same dimension, and the four convolution kernel weights are calculated by the input features. The dynamic convolution calculation process is shown in Equation 2, and the flow is shown in Figure 1B.

$$out(x) = \alpha((\partial_1 k_1 + \partial_2 k_2 + \partial_3 k_3 + \partial_4 k_4) \times x) \tag{2}$$

where α_i is the input sample-dependent weighting parameter, α is the activation function, k_i denotes each convolution kernel, × denotes the convolution operation. α_i is obtained by the four calculations in the dashed box in Figure 1B, which is shown in Equation 3.

$$\partial_i(x) = Sigmoid\ (GAP(x)R) \tag{3}$$

where R denotes the matrix that maps the input dimensions to the number of convolutional kernels. The Sigmoid function represents the weights of the four convolution kernels generated, and the GAP represents the compression of the feature layers to obtain global spatial information. Dynamic convolution increases the width and depth of the network, which improves the feature extraction capability of the algorithm by combining the information obtained from multiple convolutional kernels.

2.2. Improved FPN structure (two-branch FPN structure)

Increasing the number of FPN layers on top of the three-layer FPN structure can increase the throughput of low-sampling multiplicity features to the detection head, while increasing the transmission distance of high-sampling multiplicity features by increasing the depth of the FPN structure. To further reduce the negative impact of increased feature transmission distance, an improved FPN structure is proposed, namely, the two-branch FPN structure. This structure increases the output channels of the low-sampling multiplier features without increasing the depth of the FPN structure. The added branches are consistent with the structure and parameters of the largest branch in the original FPN, which improves the feature transfer capability of the FPN and enables the network detection head to acquire more scale features. Figure 3A shows the results of the original neck network in YOLOv4. The structure of the improved FPN based on YOLOv4 is shown in Figure 3B. The output of each of these levels of characteristics can be demonstrated more intuitively using qualitative Equations 4–7.

$$Z_4^{'} = Z_4 \tag{4}$$

$$Z_3^{'} = h_3(Z_3) \tag{5}$$

$$Z_2^{'} = h_2(Z_2, C_2) \tag{6}$$

$$Z_1^{'} = h_1(Z_1, C_1) \tag{7}$$

where $Z_1^{'}, Z_2^{'}, Z_3^{'}, Z_4^{'}$ are the four feature outputs of the two-branch FPN, h_1, h_2, and h_3 represent the three convolutional blocks A, B, and C in Figure 3B that aggregate and perform convolutional operations on the input information, Z_1, Z_2, and Z_3 represent the information inputs of these three convolutional blocks, respectively. C_2 is the target information transferred from the third dimension to the second dimension, and C_1 is the target information transferred from the second dimension to the underlying layer.

The total number of output branches is the same as that of the four-layer FPN, but the two-branch FPN is designed by parallel branching to increase the output without increasing the depth of the FPN, which can reduce the distance of feature transfer to some extent. The reduction in the distance of information transmission will certainly greatly reduce the loss of information in the transmission process, thus enhancing recognition accuracy.

2.3. Introduction of ES attention guidance module

ESNet can improve the algorithm's ability to extract valid information from small targets. As shown in Figure 1Ca, after the

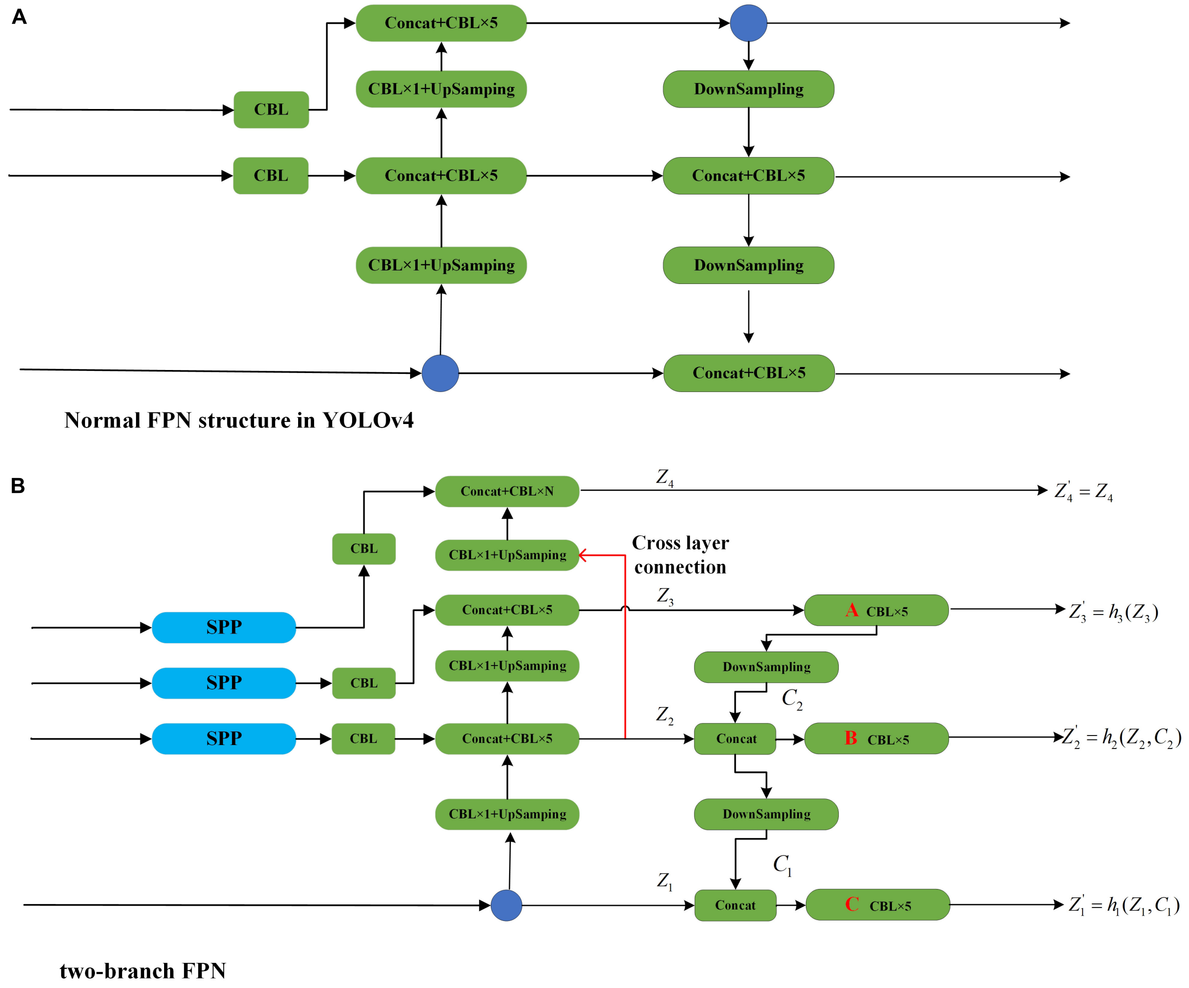


FIGURE 3
Schematic diagram of the YOLOv4 network structure based on two-branch FPN. Panel **(A)** corresponds to the FPN structure of YOLOv4, and **(B)** corresponds to the two-branch Feature Pyramid Networks (FPN) structure proposed in this paper. A, B, and C in **(B)** correspond to the three convolutional blocks at the end of the three paths, respectively.

ith module, Fi goes through a preprocessing module consisting of three convolutional layers of two 1 × 1 and one 3 × 3 and two ReLU functions to obtain the preprocessing result F_p:

$$F_p = C_{1\times1}(\partial(C_{3\times3}(\partial(C_{1\times1}(F_i))))) \tag{8}$$

∂ in this formula is the ReLU function, $C_{1\times1}$ means after a *1 × 1* convolution operation and $C_{3\times3}$ means after a *3 × 3* convolution operation.

Ineffective and redundant parameters in the network still hinder the further improvement of the network performance, so the attention-guided layer is proposed (Chen et al., 2021). This can automatically discard some unimportant attention features and dynamically adjust the weight share of multiple modules, thus improving the representational and generalization capabilities of the network.

Figure 1Cb shows the structure of Attention-guide layer. The preprocessing result F_p first increases the perceptual field through the global pooling operation and then obtains the feature information of the image through the FC, the ReLU function and the FC in turn, and finally generates the dynamic weights Wi of different modules through the softmax function with the following equation:

$$W_i = f_{agl}(F_p), (i = (1, 2)) \tag{9}$$

f_{agl} in this formula is the operation of the attention guidance layer.

The different module weights obtained through this layer will be W_1 and W_2, which are multiplied by the feature information obtained from the ECA and SGE modules, respectively, and then summed to obtain more comprehensive effective features. Finally, the input features Fi are used to obtain the feature map which based on attention mechanisms. The attention-based feature map F_{i+1} is expressed by the formula:

$$F_{i+1} = W_1 ECA(F_p) + W_2 SGE(F_p) + F_i \tag{10}$$

2.4. Real-ESRGAN algorithm

Flames often occupy less than one ten-thousandth of the field of view when shooting forest scenes at altitude or from a distance, and features are too sparse. The enhanced super-resolution generative adversarial network [ESRGAN (Wang et al., 2018)] is used to approximate the low-resolution fire samples to the high-resolution samples to achieve target feature enhancement. The algorithm is based on residual blocks, which are modified from SRRes-Net (Ledig et al., 2017) as the basic framework of the algorithm.

While Real-ESRGAN (as shown in **Figure 1D**) extends the capacity of ESRGAN into realistic recovery applications using pure synthetic data, the algorithm constructs a higher-order degradation modeling process to better simulate the realistic degradation process of images in complex situations. In addition, the algorithm employs a UNet discriminator with spectral normalization to improve the discriminative power and stabilize model training.

The classical degradation model does not cope perfectly with the complexity and variability of real-life degradation processes. For example, the original image may have been taken many years ago, and contain severe degradation problems; when the image is edited by sharpening software, it introduces overshoot and blurring artifacts, and when the image is transmitted over the network, it introduces further unpredictable compression noise. To alleviate these problems, the algorithm proposes a higher-order sharpening model. This contains multiple iterative degradation processes, which are defined as follows:

$$A(Y) = [(Y \otimes K) \downarrow_r + N]_{jpeg} \tag{11}$$

where Y is the original image, k is the blur function, $\downarrow_r$ is the downsampling factor, n is the noise and $[]_{JPEG}$ is the result compressed using the JPEG method.

Equation 12 shows the equations of a higher order degenerate model based on a first order degenerate model like Equation 11.

$$X = A^N(Y) = (A_N \cdots A_1)(Y) \tag{12}$$

where X denotes the output of the higher order degradation model and N denotes the number of steps.

In this process, each stage uses the same degradation treatment but has a different degradation super-reference.

Ringing artifacts usually appear as pseudo-edges near the sharp edges of the image; overshoot artifacts are often accompanied by ringing artifacts, which appear as jumps in the edge transition. The main reasons for these artifacts are signal bandwidth limitations and the absence of high frequencies. These artifacts usually occur when processing with sharpening algorithms, JPEG compression, etc. To solve this problem, the algorithm uses a sinc filter to simulate both of these artifacts, and the filter kernel is represented as follows:

$$k(i,j) = \frac{w_c}{2\pi\sqrt{i^2+j^2}} J_1(w_c\sqrt{i^2+j^2}) \tag{13}$$

where *(i,j)* denotes the kernel coordinate of the filter (similar to how Gaussian blurring also has such a kernel coordinate) and w_c denotes the truncation frequency. J_1 is a first order Bessel function. This equation is from Equation 6 in literature 20.

The algorithm performs this *sinc* filter processing in two places: the blurring process and the final synthesis step. The final *sinc* filter is swapped randomly with the JPEG compression to cover a larger degradation space.

2.5. Use the TensorRT framework to speed up inference

To enhance the applicability of the algorithms in this paper, we use some of the TensorRT architecture to speed up inference when the algorithms are migrated to small embedded devices based on the ARM architecture. TensorRT is NVIDIA's highly efficient inference engine, which consists of two phases: build and deployment. In the build phase, TensorRT performs several important transformations and optimizations to the Neural network graph: (1) eliminating layers of unused output to avoid unnecessary computation. (2) Fusing Convolution, Bias and ReLU layers to form a single layer, mainly vertical and horizontal layer fusion, reducing computation steps and transfer time. In the deployment phase, TensorRT runs the optimized network with minimized latency and maximized throughput. The trained weight file (.pt) is converted into an engine file (.engine) and dynamic library (.dll) *via* C language, which are deployed in the network to give the model accelerated inference. The common data structures that TensorRT can transform are INT8, INT16, INT32, FP16, FP32. The final data format chosen for this paper is the highest precision FP32.

TABLE 1 Ablation experiment results.

Number	Improved GhostNet[a]	Improved FPN[b]	ESNet[c]	AP6 (%)	FPS5
1 (YOLOv4)	–	–	–	77.57 (datum line)	48 (datum line)
2	√	–	–	70.78 (−6.79)	88 (*1.83)
3	–	√	–	83.51 (+5.94)	47 (*0.98)
4	–	–	√	82.48 (+4.91)	47 (*0.98)
5	√	√	–	75.63 (−1.94)	86 (*1.79)
6	√	–	√	74.95 (−2.62)	87 (*1.80)
7	–	√	√	86.55 (+6.98)	43 (*0.90)
8 (FireYOLO)	√	√	√	80.81 (+3.24)	84 (*1.75)

[a] GhsotNet is a lightweight neural network framework that was proposed in the literature16 and a detailed description of it in this paper can be found in 3.1.
[b] FPN stands for Feature Pyramid Networks, which was first proposed in the literature (Shen et al., 2018) and which is also an important component in the YOLOv4 neck network.
[c] This paper proposes a new attention-guiding module, the principle of which is described in 3.3 and the structure of which is shown in **Figure 1C**.
The * means multiplication sign.

3. Experimental setting

3.1. Training dataset

Several datasets related to flames [which includes FLAME (Shamsoshoara et al., 2021)] were collected and divided into two parts. The datasets used for training the model include single-flame, multi-flame, indoor fire, forest fire, and complex background fire scenarios, with a total of 23,982 images. During the training of the model, 23,982 images were divided into 18,653 training image sets and 5,329 validation image sets in a ratio of 7:2. Before input to the training framework, there was no imposition on the image size, and after input to the training framework, the images of various sizes were scaled uniformly to 416 × 416 by the algorithm. while the datasets used for the comparison experiments in the subsequent experiments of the paper were all UAV remote sensing images of high altitude areas, 7,752 images were used in this dataset, which were divided into two categories according to their types: The first category is the images containing fire, named FIRE[1] type, with 6,331 images; and the second category is the images without fire, named NOFIRE[2] type, with 1,421 images. The algorithm runs with no size requirement for the image being inspected.

3.2. Model building and training

Considering that the subsequent verification process of the algorithm in this paper involves many comparison experiments, some complex algorithms will be applied in these comparison experiments, and most of these algorithms cannot be run on embedded devices. In the principle of controlling variables, the platform of the pre-contrast validation experiments in this paper is unified with the model training platform. The platform used for training and experiments is CUDA 11.2 CUDNN v8.2.1, the deep learning environment is Tensorflow 2.5, the programming language is Python 3.9, and the system is Ubuntu 18.04.

There is no pre-training process for the FireYOLO model, and the model is trained directly from scratch with the following training hyperparameters settings: epochs for training is set to 1,000; batchsize is set to 64 and subdivisions is set to 1.

3.3. 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. The test set is divided into two categories, positive samples, and negative samples.

This paper uses the accuracy (AR)[3], Recall[4] [detection rate (DR)], False Accept Rate [5] (FAR), Average Precision (AP)[6], and running frame rate FPS[7] as the evaluation indicators of the algorithm. The formula for calculating the above metrics is shown in Equations 14–19, where Equation 17 means that a graph is constructed with accuracy as the vertical coordinate and recall as the horizontal coordinate and then the area under the curve in that graph is calculated by the principle of calculus.

$$Recall(orDR) = \frac{TP}{TP + FN} \tag{14}$$

$$FAR = \frac{FP}{FP + TN} \tag{15}$$

$$FN_{rate} = \frac{FN}{FN + TP} \tag{16}$$

$$AP = \int_0^1 P(r)dr \tag{17}$$

$$NFAR = 1 - \frac{FP}{FP + TN} \tag{18}$$

$$AR = \frac{TN + TP}{TN + FN + FP + TP} \tag{19}$$

4. Results and discussion

4.1. Experiment on the recognition effect of FireYOLO

To be able to verify the effectiveness of the three components of the FireYOLO improvement and whether there is some conflict and exclusion between these improvements, we conducted ablation experiments on FireYOLO (Table 1). The ablation experiments are similar to the control variables approach in that when only one of the three components is changed, we can analyse the effectiveness of this component improvement by comparing the experimental data before and after the change, as in Experiments 2–4; by changing two or three of the three components, we can verify whether these improved components can collaborate with each other to further improve the performance of the algorithm, as in Experiments 5–8 in Table 1. By comparing Experiments 1 and 2, we found that the introduction of dynamic convolutional GhostNet as the head structure of the FireYOLO algorithm resulted in a significant increase in FPS; by comparing Experiments 1 and 3, we found that the two-branch FPN structure resulted in

[1] The FIRE type indicates the presence of at least one fire phenomenon in the UAV remote sensing image.

[2] NOFIRE says there are no drone remote sensing images of fires occurring.

[3] AR stands for Accuracy, which is calculated as shown in 19.

[4] Recall (DR) refers to the proportion of successful predictions of the algorithm among all true positive classes, and it is calculated as shown in 14.

[5] The full name of FAR is False Accept Rate, and its calculation formula is Equation 15.

[6] AP stands for Average Precision and is calculated according to Equation 17. AP@0.5 is calculated in the same way as AP. The difference is that AP@0.5 requires an IOU greater than or equal to 0.5 in order for the algorithm to be counted as detecting the target.

[7] FPS is how many images per second the target network can process, which is simply understood as how often the images are refreshed. The faster the algorithm runs, the higher the FPS.

a significant increase in AP and almost no decrease in FPS. The comparison of Experiment 1 and Experiment 4 shows that ESNet can significantly increase the AP of the algorithm with almost no further decrease in FPS. By comparing Experiment 1, Experiment 2, and Experiment 5, we can see that the AP of Experiment 5 is much improved compared to Experiment 2 and the FPS is also much improved compared to Experiment 1, which proves that the improved GhostNet structure can improve the performance of the algorithm together with the two-branch FPN structure. Similarly, the analysis of Experiments 1, 5, 6, 7, and 8 shows that all three improvements can jointly improve the overall performance of the algorithm.

In this paper, six common deep learning image recognition algorithms were used for fire detection, and the final comparison results were shown in Table 2 below. And Supplementary Figure 1 was drawn from Table 2, according to the trend of this line graph, the algorithm in this paper could achieve the best balance between recognition speed and accuracy. FireYOLO was only slightly slower than YOLOv5, but its accuracy was significantly higher than YOLOv5. In this paper, the confusion matrix (Figure 4) is used to further compare the performance of the algorithms. The number of six regions corresponding to NOFIRE^2 in the horizontal coordinate of Figure 4A represents the number of images missed by the six algorithms when recognizing images of type FIRE^1 plus the number of false detections (e.g., the algorithm identifies an image with a fire occurring, but the detection result does not frame out the fire part but other non-fire parts, a situation that is typical of false detection.); the number of six regions in the second column of Figure 4B represents the number of NOFIRE^2 type images identified by the six algorithms without detecting a flame; the number of attributions for TP, FN, FP, and TN in Figure 4C is based on the data in Figures 4A, B. From Figure 4C, we can see that the number of FN (number of missed images) is much smaller than the number of FP (number of false detections), with FP having a relatively large value. However, the TP and TN of the algorithm in this paper are still higher than the other algorithms, and the FN and FP are still lower than the other algorithms, which further illustrates the advantages of FireYOLO.

Figure 5 showed heatmaps for the various algorithms, making the inference process easy to observe. Comparing the heatmap results for rows 1–6 with row 8, it could be seen that row 8 has the highest focused on the flame target and the strongest aggregation of attention (the area of the dark red distribution almost matched the area of the flame the best), with a very clear demarcation line between the flame and the forest background and a less clear green dispersion. Comparing rows 7 and 8, it could be seen that the green dispersion in row 7 was very strong and the demarcation line between the flame and forest background was not clear enough, but with the introduction of ESNet the demarcation line became clear. This further demonstrated the ability of ESNet to guide valid information about small targets.

TABLE 2 Comparison of different methods.

Method	Params[a] (M)	AP[6] (%)	AP@0.5[6] (%)	FPS[5]
A. Faster-RCNN	108	55.56	46.03	20
B. SSD	90.57	56.41	49.17	60
C. YOLOv3	234.67	71.12	63.13	51
D.YOLOv4	243.91	77.57	68.61	48
E. YOLOv5x	27	74.13	65.13	99
F. FireYOLO	50.71	80.81	70.33	66

[a] Params indicates the model size of the algorithm in M.

4.2. Validation of Real-ESRGAN for resolution enhancement of fire

The image super-resolution approach aims to recover detailed SR images from the corresponding LR images, and the Real-ESRGAN network was experimentally compared with FSRCNN (Dong et al., 2016) and ESRGAN for the resolution enhancement of small-size flames and pairs to verify its effectiveness. To evaluate the quality of the generated SR images, the RMSE, NRMSE, SSIM, PSNR, and Entropy of the test images were compared, and the results are shown in Table 3. Figure 6 shows the effects of the three algorithms after enhancing the pixels of the small-size fire image. From this figure we could see that Real-ESRGAN was a little more capable of pixel enhancement. As shown in Table 3, both the SSIM and PSNR image quality metrics of Real-ESRGAN were higher than those of other hyper-segmentation networks.

4.3. Overall performance of FireYOLO and Real-ESRGAN combined

4.3.1. The influence of objective factors such as shading or changes in light on the algorithms

In order to discuss the practicality of our fire detection algorithm and its adaptability to the characteristics of forests with many grass-like occlusions, in this section we discussed the effectiveness of our algorithm for detecting flames when they were occluded.

We experimentally verified multiple performances by collecting 2,000 random photos from the 6,331 photos. Table 4A shows the recognition results of multiple algorithms in the face of occluded scenes, while Supplementary Figure 2 was a line graph drawn from the data in this table, and according to the trend of this line graph our algorithm was the best in AP, Recall, and AR metrics, and the FPS was at the average level among these algorithms, but also very fast. So all together this made our algorithm still the best in overall performance. Figure 8A showed the specific recognition results of these algorithms. From these result plots we could visually see that the algorithm in this paper identifies all the obscured targets, while the other algorithms all had targets that were missed.

Considering that illumination can have a significant impact on the target detection algorithm, this section focuses on the degree of adaptation of our algorithm to changes in illumination. We selected 1,000 photos from the 6,331 images for each of the three types of light intensity: sunny, cloudy, and dark. The various detection characteristics of the seven algorithms under sunny, cloudy and dark conditions were given in Figure 4B. The four line graphs in Figure 7 visualize the data in Table 4B. Figures 7A–C represented the trends of the seven algorithms regarding the three evaluation metrics mentioned above under the three lighting conditions, and

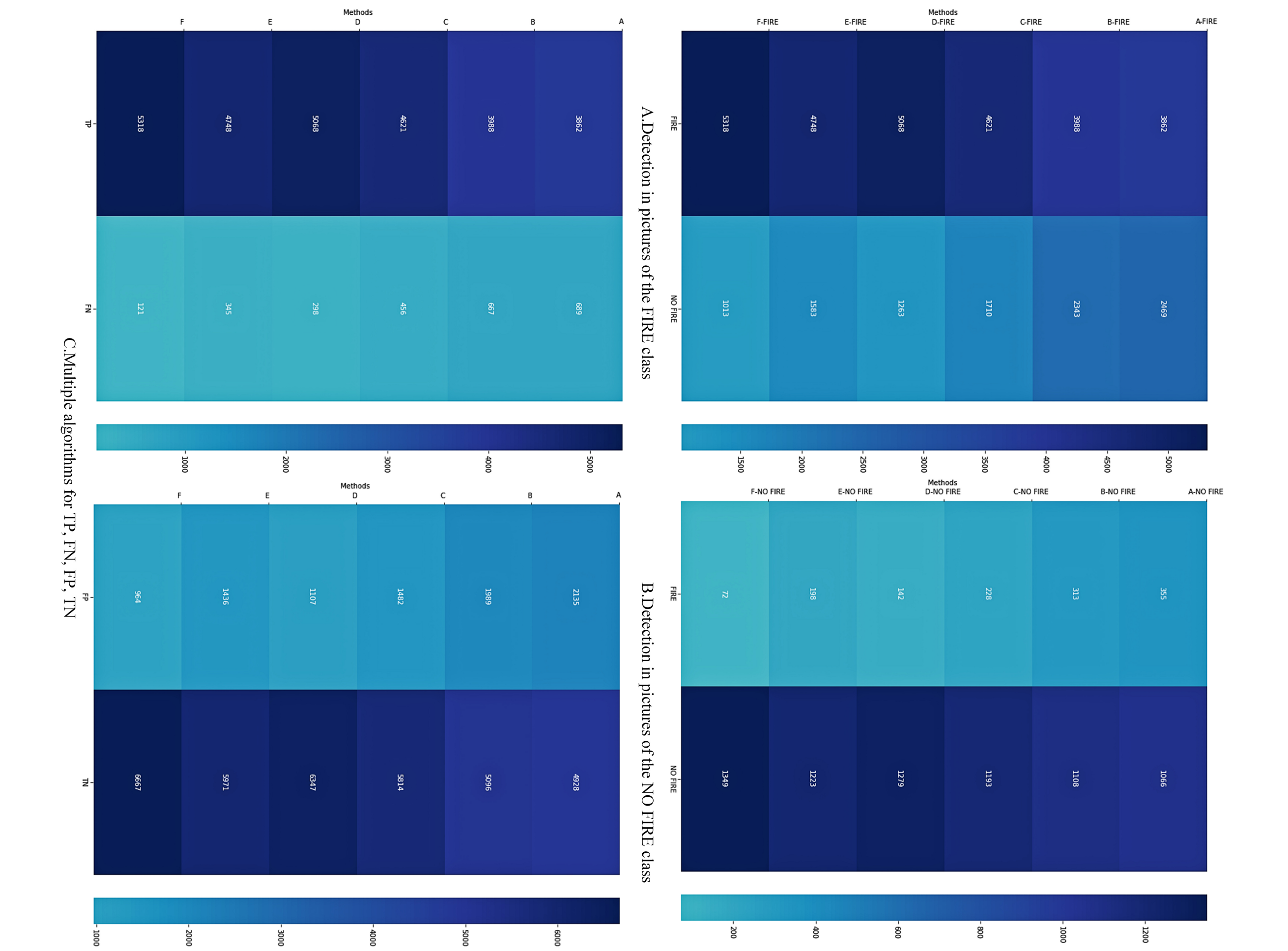


FIGURE 4
On the confusion matrix of the detection results of the six algorithms **(A)** shows the results of the six algorithms for the FIRE^1 class of images, where the horizontal coordinate FIRE indicates that the algorithm correctly identified the flame and NOFIRE indicates that the algorithm did not correctly identify the flame, possibly because the algorithm missed or misidentified it; **(B)** shows the results of the six algorithms for the NOFIRE^2 class of images, where the horizontal coordinate FIRE indicates that the algorithm misidentified the flame and NOFIRE indicates that the algorithm did not identify the flame; **(C)** shows the confusion matrix of the six algorithms for the test dataset regarding the distribution of the number of attributes TP, FN, FP, and TN. The darker the color of a region, the greater the number of images corresponding to that region.

it could be found that the algorithm in this paper was far ahead in these three metrics under all three lighting conditions; Figure 7D represented the trends of the evaluation metrics of the algorithm (G) in this paper when the lighting changed, and we found that the fire identification in darkness was better than the other two lighting conditions. Figure 8B showed the graphs of the recognition effects of multiple algorithms regarding the three lighting conditions, from which it could also be visualized that the G algorithm had the best recognition effect under the three lighting conditions, followed by FireYOLO.

4.3.2. Comparison with other fire detection methods

To verified the generality of our proposed algorithm for the environment, we compared it with current state-of-the-art fire detection algorithms on publicly available datasets. We compared the results of Muhammad et al. (2018a,b), Chaoxia et al. (2020), Pan et al. (2020), and Our algorithm under the BoWFire (Chino et al., 2015) dataset. The BoWFire dataset was derived from real fire and urban fire scenarios. The specific data of this experiment were shown in Table 5 below.

In the analysis in this section we focus on the A–D algorithm in Table 5 with the four evaluation metrics of FireYOLO (E) proposed in this paper. In the above we have experimentally concluded that FireYOLO has very few missed images but many false positives, in other words a low FN but extremely high FP values, and by analysing Equations 14, 15, 19 we conclude that FireYOLO will have a high FAR and low AR and DR. However, this does not mean that FireYOLO is very inaccurate, as the experimental data in Table 5E show that FireYOLO's accuracy can be at the top of all these algorithms, but it is really not good enough.

We then move on to analyze the reasons for the change in these four evaluation metrics when upgrading from the E to the F algorithm: the final detection results for the FIRE^1 type dataset are divided into three types: correctly detected flames, missed flames and misdetected flames (there will also be images that are both misdetected and missed), then when the algorithm is upgraded from E to F, in addition to the number of misdetected images being

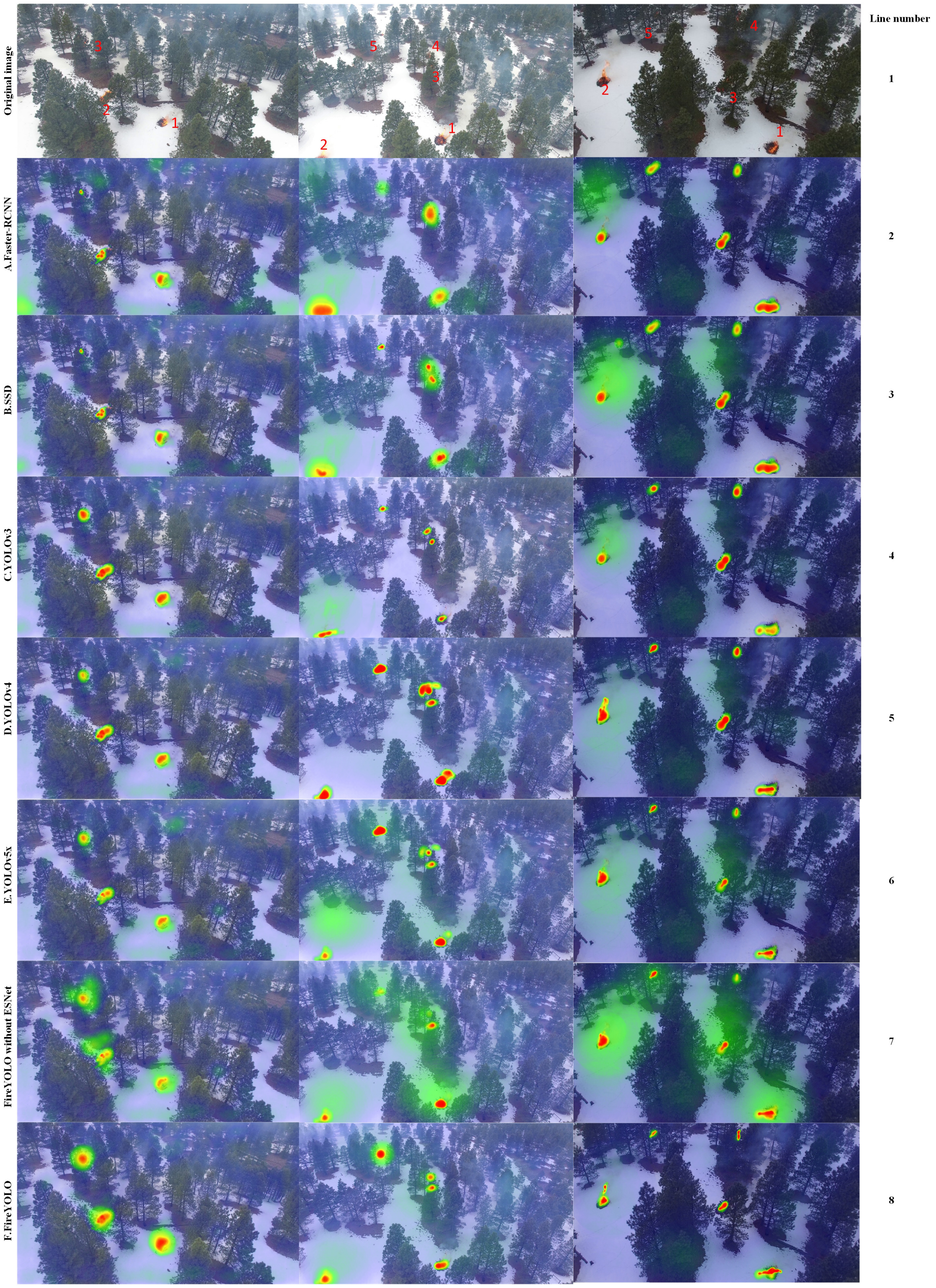


FIGURE 5
Comparison of FireYOLO with other advanced target detection algorithms. The shade of color indicates how much attention the algorithm is paying to the image, with a darker color in a region indicating that the algorithm is paying more attention to that region.

greatly reduced, the number of missed The detection results for the NOFIRE^2 type images fall into two categories: The detection results for missed NOFIRE^2 type images are divided into two categories: flames detected (false detection) and no flames detected (correct). When the algorithm is upgraded from E to F, the number of images with flames detected will be greatly reduced, while the number of missed flames will be greatly increased. When upgrading from E to F, the number of images with detected flames will be greatly

TABLE 3 Comparison of different methods.

Model	RMSE[a]	NRMSE[b]	SSIM[c]	PSNR[d] /dB	Entropy[e] /bit
FSRCNN	27.165	0.567	0.776	17.167	1.675
ESRGAN	4.835	0.045	0.867	36.443	6.672
Real-ESRGAN	2.865	0.023	0.967	42.443	8.672

[a] RMSE is called Root Mean Square Error, which is calculated as the square of the difference between the true value and the predicted value, then summed up and averaged, and finally opened to the root. The smaller the value, the higher the image quality.
[b] NRMSE is called Normalized Root Mean Square Error, which means that the value of RMSE becomes between 0 and 1, and the smaller its value, the higher the image quality.
[c] SSIM is called structural similarity, and it ranges from (0,1), and the larger the value, the better the quality of the image. When two images are exactly the same, SSIM = 1 at this time.
[d] PSNR is called Peak Signal to Noise Ratio, which is the ratio of the energy of the peak signal to the average energy of the noise, the larger its value, the higher the picture quality.
[e] Entropy is mainly a measure of how much information an image contains, and a higher value means more information and better image quality.

reduced, while the number of images with undetected flames will be greatly increased. The final combination of these changes will result in a dramatic increase in AR, DR and NFAR as well as a dramatic decrease in FAR.

When we compared the differences between the four evaluation criteria of the A–D and F algorithms, we found that when these algorithms were applied to the BoWFire dataset, the algorithm in this paper achieved the highest AR (95.6%) and NFAR (97.7%), and the lowest FAR (2. 3%), these excellent metrics were mainly due to the advantages of the structure of the algorithm in this paper: FireYOLO possessed the characteristics of low computational effort and low false detection rates for different scales of flame detection; The use of the Real-ESRGAN algorithm to improve the local sharpness of the image allowed the secondary recognition of FireYOLO to significantly reduce the false detection rate, which also led to a significant increase in the overall accuracy of the algorithm. However, the inability of the algorithm in this paper to reduce the false detection rate of FireYOLO made the final DR of the algorithm (95.8%) no higher than that of the A algorithm (97.5%). However, in terms of overall algorithm performance, F is still much better than A.

4.3.3. The algorithm in this paper runs on small embedded devices (Jetson NX)

To verify the adaptability of our algorithm to some embedded platforms with smaller computing power, we migrated the algorithm to the Jetson Xavier NX after accelerating it through the TensorRT framework, and the migrated algorithm was compared with several other algorithms running on the Jetson Xavier NX, and the final results were shown in Table 6. Supplementary Figure 3 showed the trend of the four evaluation metrics about these algorithms drawn from the data in Table 6. The A–F algorithm in this table is almost identical to the AP trend in Table 2, while AR and Recall also almost outperform the A–E algorithm, mainly due to the fact that we have included many structures in YOLOv4 that enhance effective information extraction (i.e., ESNet and two-branch FPN structures), and the algorithms in this paper also run at the top of these algorithms, only 7 FPS lower than YOLOv5x. This is mainly due to our choice to use the Ghost-Net structure instead of the original CSPDarknet structure, which significantly reduces the overall computational effort of the algorithm. The reasons for the change in trend between the F and G algorithms in Table 6 are largely similar to the reasons for the change in trend between the E and F

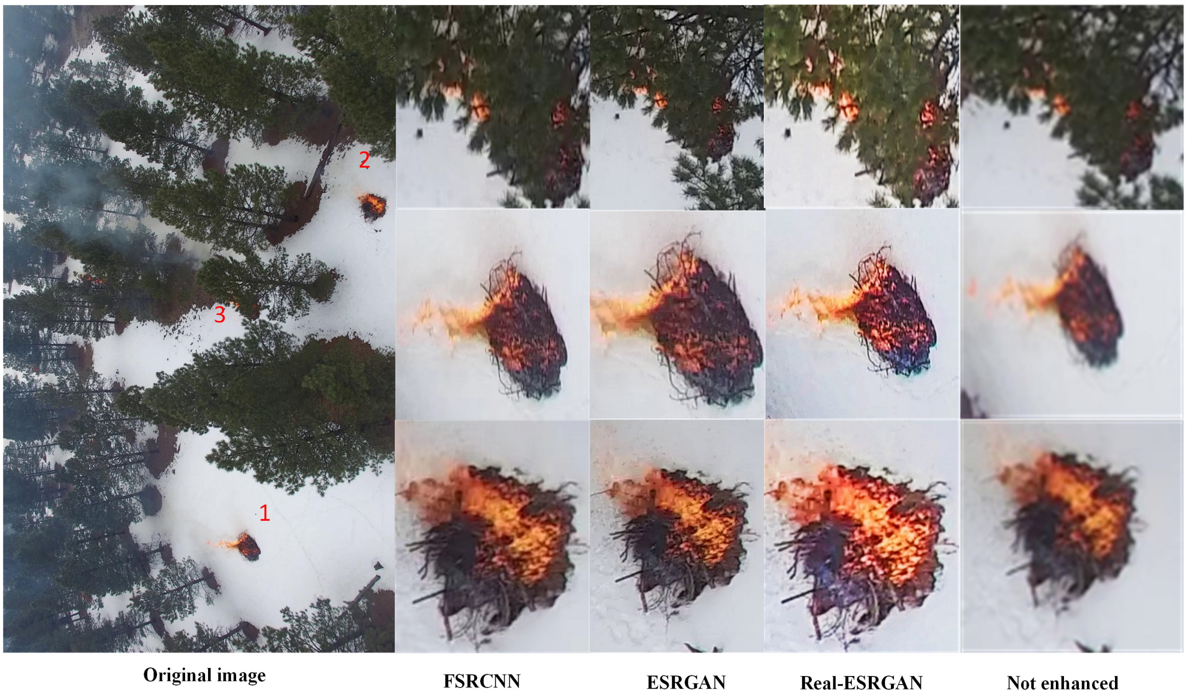
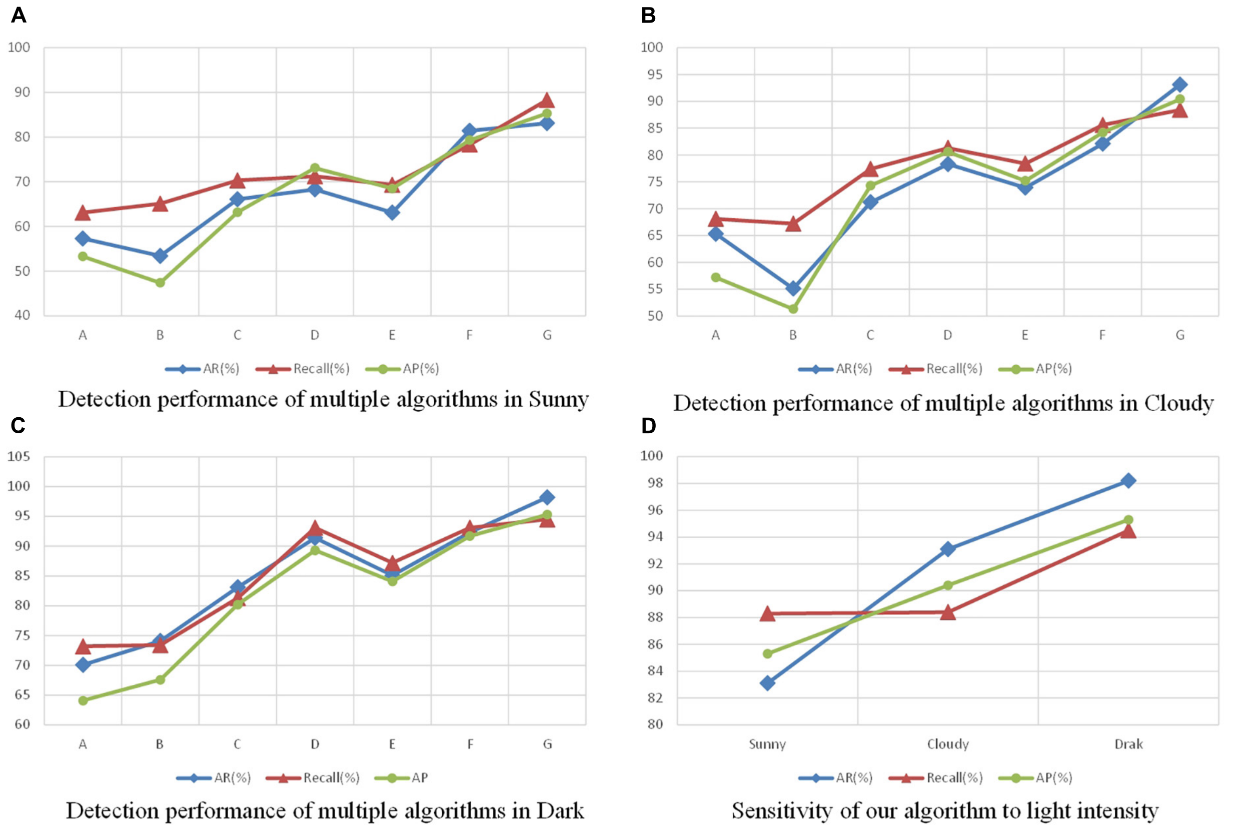


FIGURE 6
Algorithms used to enhance the rendering of image pixels.

TABLE 4 Combined performance of multiple algorithms under changing light or shading conditions.

Methods	A. Obscuration issues				B. Light changes								
					Sunny			Cloudy			Dark		
	AR^3	Recall^4	AP^6	FPS^5	AR	Recall	AP	AR	Recall	AP	AR	Recall	AP
A. Faster-RCNN	55.4	43.4	48.1	20	57.3	63.1	53.3	65.3	68.1	57.18	70.1	73.2	64.1
B. SSD	43.4	45.7	46.5	60	53.4	65.1	47.4	55.1	67.2	51.3	74.1	73.4	67.6
C. YOLOv3	69.1	56.8	64.2	51	66.1	70.3	63.2	71.2	77.4	74.3	83.1	81.3	80.2
D.YOLOv4	70.3	65.3	66.6	48	68.3	71.2	73.1	78.3	81.3	80.6	91.4	93.1	89.3
E. YOLOv5x	67.1	59.4	63.4	101	63.1	69.3	68.5	73.9	78.4	75.2	85.1	87.2	84.1
F. FireYOLO	76.3	70.3	72.7	66	81.4	78.3	79.3	82.1	85.6	84.2	92.3	93.1	91.7
G. Our methods	86.7	80.4	85.3	46	83.1	88.3	85.3	93.1	88.4	90.4	98.2	94.5	95.3



Detection performance of multiple algorithms in Sunny

Detection performance of multiple algorithms in Cloudy

Detection performance of multiple algorithms in Dark

Sensitivity of our algorithm to light intensity

FIGURE 7
Line graph of the accuracy of various algorithms under different lighting conditions. Panels **(A–C)** show the trends of the evaluation metrics for each of the seven algorithms under the three different lighting conditions. Panel **(D)** represents the trend of the evaluation metrics of the algorithm (G) in this paper under the variation of light.

algorithms in Table 5, mainly because the introduction of the Real-ESRGAN algorithm reduced the number of images that were incorrectly detected during the recognition process of the FireYOLO algorithm.

The results of the algorithm's run are shown in Figure 9. From this figure, we first looked at the first column of remotely sensed images and we found that Algorithms A–D all more or less missed the flames obscured by the trees, while Algorithms E–G all detected the flames on this remotely sensed image and Algorithm G identified the flames with the highest overall confidence. We looked at the second column of remote sensing images and found that the second column had a smaller flame area than the first column, which was not detected by any of Algorithms A–E, but Algorithms F and G detected all the fires. The third column of remote sensing images also showed a similar situation to the second column, with Algorithms A–E all having missed detections and Algorithm D also having false detections, while Algorithms F and G had no missed detections and G identified the flames with much higher confidence than F. We then proceeded to analyse the remotely sensed images in the fourth column and found that all algorithms failed to detect

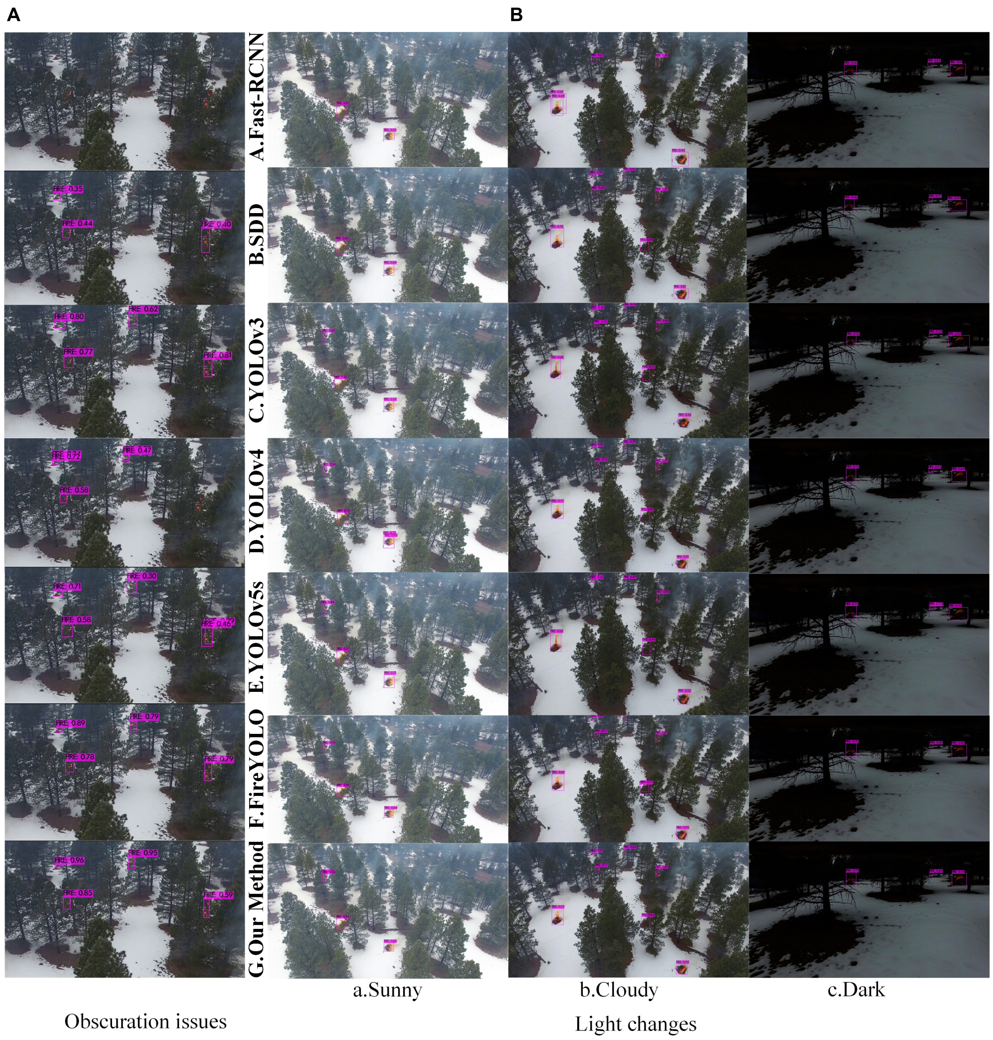


FIGURE 8
Plot of the results of various algorithms in different lighting conditions or with or without shading. Panel **(A)** represents the detection results of the seven algorithms when an occlusion situation occurs and **(B)** represents the detection results of the seven algorithms under three lighting conditions.

the two flames in the red circle that were about to go out, mainly because they were not red enough and did not differ much from the background color of the surrounding forest. However, when we analysed the set of flames detected by all algorithms, we found that G had the highest confidence level, followed by F. As can be seen from the four images in Figure 9, the G algorithm had the lowest rate of missed and false detections, followed by the FireYOLO (F) algorithm proposed in this paper.

TABLE 5 Comparison results of multiple algorithms.

Dataset	References	AR[3] (%)	DR[4] (%)	FAR[5] (%)	NFAR[a] (%)
BoWFire	A. Muhammad et al., 2018b	89.8	97.5	18.7	81.3
	B. Muhammad et al., 2018a	92.0	93.3	9.3	90.7
	C. Chaoxia et al., 2020	93.4	92.4	5.6	94.4
	D. Pan et al., 2020	93.4	91.6	4.7	95.3
	E. FireYOLO	92.7	93.7	7.8	92.2
	F. Our method	95.6	95.8	2.3	97.7

[a] NFAR is the non-false detection rate, which is calculated by Equation 18.

TABLE 6 Performance of various algorithms on Jetson Xavier NX.

Composition of algorithm	AR[3] (%)	Recall[4] (%)	AP[6] (%)	FPS[5]
A. Faster-RCNN	61.30	52.18	54.67	0.25
B. SSD	59.12	53.29	57.38	2.5
C. YOLOv3 (+TensorRT)	69.13	73.80	70.18	31.12
D. YOLOv4 (+TensorRT)	73.22	79.40	77.09	28.03
E. YOLOv5x (+TensorRT)	71.31	73.31	73.13	40.42
F. FireYOLO (+TensorRT)	76.91	81.45	79.14	33.12
G. Our method (+TensorRT)	95.11	86.61	94.22	20.67

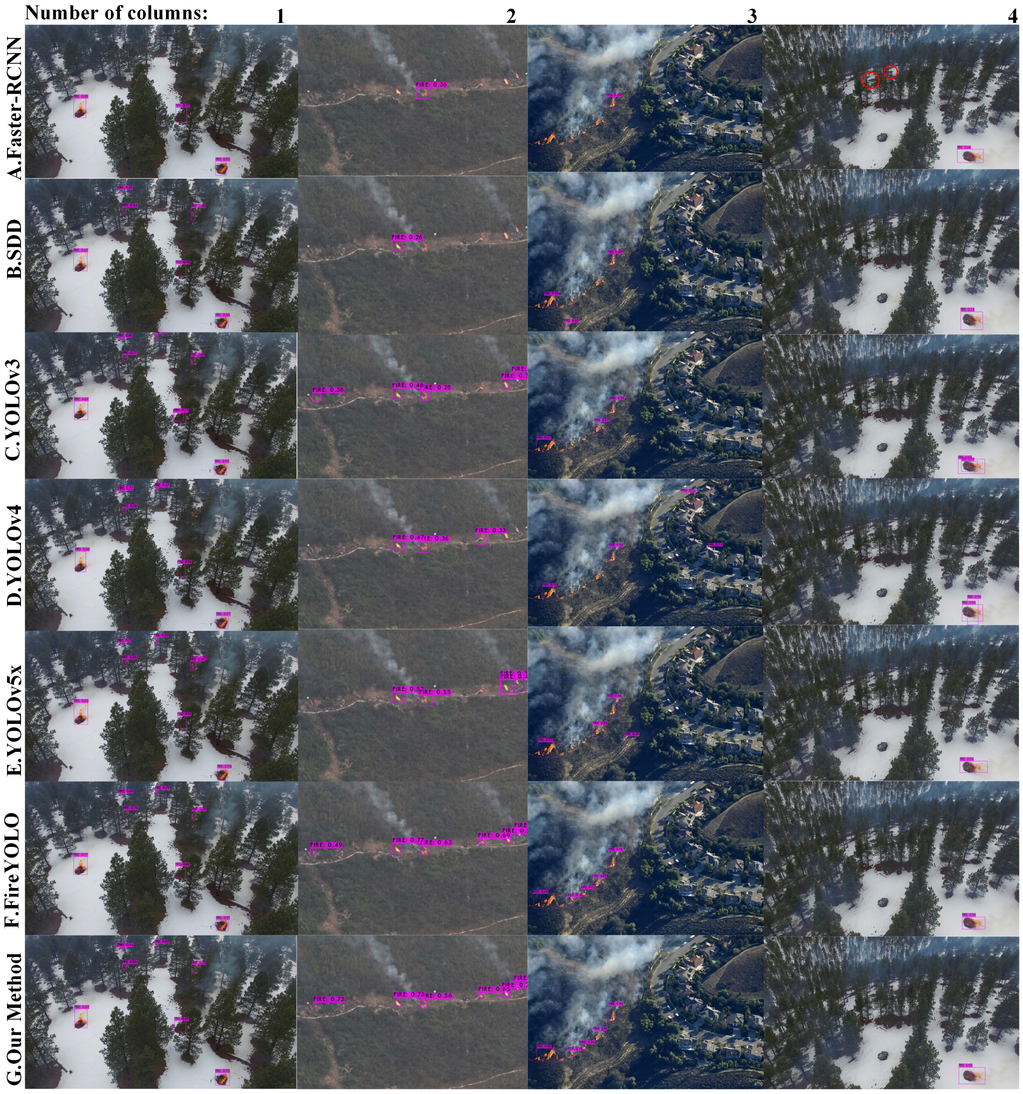


FIGURE 9
Running results of algorithm. The first to third columns of this figure do not show any missed detections for the algorithms in this paper, while the fourth column shows that there are missed detections for the algorithms in this paper. Two red circles indicate missed flames.

5. Conclusion

A lightweight two-step small-scale fire detection method based on FireYOLO and Real-ESRGAN is proposed in this paper. Based on the results of our experiments, we have drawn four conclusions:

(1) The proposed two-branch FPN and ESNet can effectively improve the small target information extraction capability of FireYOLO while reducing the information conduction loss. Meanwhile, using GhostNet with dynamic convolution introduced as the backbone network of FireYOLO can significantly reduce computation, and thus increased the efficiency of the algorithm. The two-branch FPN, ESNet and GhostNet with dynamic convolution can work together to improve the performance of FireYOLO, and there is no exclusion between them;

(2) The FireYOLO algorithm does have a very low miss detection rate but a high false detection rate.

(3) In order to reduce the false detection rate of the FireYOLO algorithm, this paper introduces the Real-ESRGAN algorithm, which does significantly reduce the final false detection rate and improves the accuracy of the algorithm;

(4) Our algorithm combines two algorithms, FireYOLO and Real-ESRGAN. These two algorithms work in concert with each other, which makes the algorithm in this paper achieve extremely high accuracy, inference speed and strong anti-interference capability on PC side, surpassing all other algorithms. Also, through the above experiments we have found that the algorithm of this paper can still achieve 94.22% AP when deployed on embedded devices, which is much higher than other algorithms, and although the speed of this paper's algorithm is slightly lower than that of YOLOv5+TensorRT, the frame rate of this paper's algorithm has reached 20.67 FPS, which is fast enough to achieve almost no latency. In summary, the algorithm in this paper perfectly achieves a breakthrough in both inference speed and recognition accuracy, and has good application prospects.

Although this algorithm has its advantages, it still has its shortcomings: the collaboration between FireYOLO and Real-ESRGAN does not reduce the rate of FireYOLO misses, which means that the images missed by FireYOLO cannot be detected by this algorithm in the end. Even though FireYOLO's miss rate has reached a very low level, I still need to find a solution to further improve the accuracy of this algorithm.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

HZ and YL: conceptualization, methodology, software, and investigation. SD and YW: validation and writing—review and editing. HC and QZ: formal analysis. YL: resources, data curation, project administration, and funding acquisition. HZ: writing—original draft preparation and visualization. YL and SD: supervision. All authors have read and agreed to the published version of the manuscript.

Funding

This work was received support from 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).

Acknowledgments

We thank all the reviewers, and the preliminary and academic editors.

Conflict of interest

QZ was employed by Zhejiang Dahua Technology Co., Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/ffgc.2023.1134942/full#supplementary-material

References

Barbosa, M. L. F., Delgado, R. C., Teodoro, P. E., Pereira, M. G., Correia, T. P., de Mendonça, B. A. F., et al. (2018). Occurrence of fire foci under different land uses in the State of Amazonas during the 2005 drought. *Environ. Dev. Sustain.* 21, 2707–2720. doi: 10.1007/s10668-018-0157-4

Bochkovskiy, A., Wang, C. Y., and Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv* [Preprint]. doi: 10.48550/arXiv.2004.10934

Bouabdellah, K., Noureddine, H., and Larbi, S. (2013). Using wireless sensor networks for reliable forest fires detection. *Procedia Comput. Sci.* 19, 794–801. doi: 10.1016/j.procs.2013.06.104

Brancalion, P. H. S., Niamir, A., Broadbent, E., Crouzeilles, R., Barros, F. S. M., Almeyda Zambrano, A. M., et al. (2019). Global restoration opportunities in tropical rainforest landscapes. *Sci. Adv.* 5:eaav3223. doi: 10.1126/sciadv.aav3223

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

Chen, H., Gu, J., and Zhang, Z. (2021). Attention in attention network for image super-resolution. *arXiv* [Preprint] arXiv:2104.09497.

Chino, D. Y., Avalhais, L. P., Rodrigues, J. F., and Traina, A. J. (2015). "Bowfire: Detection of fire in still images by integrating pixel color and texture analysis," in *Proceedings of the 28th SIBGRAPI conference on graphics, patterns and images*, (Salvador), 95–102. doi: 10.1109/SIBGRAPI.2015.19

Cui, F. (2020). Deployment and integration of smart sensors with IoT devices detecting fire disasters in huge forest environment. *Comput. Commun.* 150, 818–827. doi: 10.1016/j.comcom.2019.11.051

De Frenne, P., Zellweger, F., Rodríguez-Sánchez, F., Scheffers, B. R., Hylander, K., Luoto, M., et al. (2019). Global buffering of temperatures under forest canopies. *Nat. Ecol. Evol.* 3, 744–749. doi: 10.1038/s41559-019-0842-1

de Santana, R. O., Delgado, R. C., and Schiavetti, A. (2021). Modeling susceptibility to forest fires in the Central Corridor of the Atlantic Forest using the frequency ratio method. *J. Environ. Manag.* 296:113343. doi: 10.1016/j.jenvman.2021.113343

Dong, C., Loy, C. C., and Tang, X. (2016). Accelerating the super-resolution convolutional neural network. *arXiv* [Preprint] arXiv:1608.00367.

Han, K., Wang, Y. H., Tian, Q., Guo, J., Xu, C., and Xu, C. (2020). "Ghostnet: More features from cheap operations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (Seattle, WA), 1580–1589. doi: 10.1109/CVPR42600.2020.00165

Han, X. F., Jin, J. S., Wang, M. J., Jiang, W., Gao, L., and Xiao, L. P. (2017). Video fire detection based on Gaussian Mixture Model and multicolor features. *Signal Image Video Process.* 11, 1419–1425. doi: 10.1007/s11760-017-1102-y

Harkat, H., Nascimento, J., and Bernardino, A. (2020). Fire segmentation using a DeepLabv3+ architecture. Image and Signal Processing for Remote Sensing XXVI. *Int. Soc. Opt. Photonics Proc. SPIE* 11533, 134–145. doi: 10.1117/12.2573902

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Spatial Pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 1904–1916. doi: 10.1109/TPAMI.2015.2389824

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE conference on computer vision and pattern recognition (CVPR)*, (Cairo), 770–778. doi: 10.1109/CVPR.2016.90

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Commun. ACM.* 60, 84–90. doi: 10.1145/3065386

Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. *arXiv* [Preprint] arXiv:1609.04802.

Li, P., and Zhao, W. (2020). Image fire detection algorithms based on convolutional neural networks. *Case Stud. Therm. Eng.* 19:100625. doi: 10.1016/j.csite.2020.100625

Li, X., Hu, X. L., and Yang, J. (2019). Spatial group-wise enhance: Improving semantic feature learning in convolutional networks. *arXiv* [Preprint] arXiv:1905.09646.

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). "Feature pyramid networks for object detection," in *Proceedings of the 30th IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, (Honolulu, HI), 936–944. doi: 10.1109/CVPR.2017.106

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., et al. (2016). "Ssd: Single shot multibox detector," in *Proceedings of the European conference on computer vision*, (Amsterdam), 21–37. doi: 10.48550/arXiv.1512.02325

Lizundia-Loiola, J., Pettinari, M. L., and Chuvieco, E. (2020). Temporal anomalies in burned area trends: Satellite estimations of the Amazonian 2019 fire crisis. *Remote Sens.* 12:151. doi: 10.3390/rs12010151

Luo, Y., Zhao, L., Liu, P., and Huang, D. (2018). Fire smoke detection algorithm based on motion characteristic and convolutional neural networks. *Multimed. Tools Appl.* 77, 15075–15092. doi: 10.1007/s11042-017-5090-2

Mitchard, E. T. A. (2018). The tropical forest carbon cycle and climate change. *Nature* 559, 527–534. doi: 10.1038/s41586-018-0300-2

Muhammad, K., Ahmad, J., Lv, Z., Bellavista, P., Yang, P., and Baik, S. W. (2018a). Efficient deep CNN-based fire detection and localization in video surveillance applications. *IEEE Trans. Syst. Man Cybern. Syst.* 49, 1419–1434. doi: 10.1109/TSMC.2018.2830099

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

Pan, H., Badawi, D., and Cetin, A. E. (2020). Computationally efficient wildfire detection method using a deep convolutional network pruned via fourier analysis. *Sensors* 20:2891. doi: 10.3390/s20102891

Redmon, J., and Farhadi, A. (2017). "YOLO9000: Better, faster, stronger," in *Proceedings of the 2017 IEEE conference on computer vision and pattern recognition*, (Honolulu, HI), 7263–7271. doi: 10.1109/CVPR.2017.690

Redmon, J., and Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv* [Preprint]. doi: 10.48550/arXiv.1804.02767

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: Unified, real-time object detection," in *Proceedings of the 2016 IEEE conference on computer vision and pattern recognition (CVPR)*, (Las Vegas, NV), 779–788. doi: 10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Sarwar, B., Bajwa, I. S., Jamil, N., Ramzan, S., and Sarwar, N. (2019). An intelligent fire warning application using iot and an adaptive neuro-fuzzy inference system. *Sensors* 19:3150. doi: 10.3390/s19143150

Sasmita, E. S., Rosmiati, M., and Rizal, M. F. (2018). "Integrating forest fire detection with wireless sensor network based on long range radio," in *Proceedings of the 2018 international conference on control, electronics, renewable energy and communications (ICCEREC)*, (Bandung), 222–225. doi: 10.1109/ICCEREC.2018.8711991

Shamsoshoara, A., Afghah, F., Razi, A., Zheng, L., Fulé, P. Z., and Blasch, E. (2021). Aerial imagery pile burn detection using deep learning: The FLAME dataset. *Comput. Netw.* 193:108001. doi: 10.1016/j.comnet.2021.108001

Shen, D., Chen, X., Nguyen, M., and Yan, W. Q. (2018). "Flame detection using deep learning," in *Proceedings of the 2018 4th international conference on control, automation and robotics (ICCAR)*, (Auckland), 416–420. doi: 10.1109/ICCAR.2018.8384711

Simonyan, K., and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *arXiv* [Preprint]. doi: 10.48550/arXiv.1409.1556

Szegedy, C. (2015). "Going deeper with convolutions," in *Proceedings of the 2015 IEEE conference on computer vision and pattern recognition (CVPR)*, (Boston, MA). doi: 10.1109/CVPR.2015.7298594

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., and Hu, Q. (2020). "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proceedings of the 2020 IEEE/CVF conference on computer vision and pattern recognition*, (Seattle, WA), 13–19. doi: 10.1109/CVPR42600.2020.01155

Wang, X., Xie, L., Dong, C., and Shan, Y. (2021). "Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data," in *Proceedings of the 2021 IEEE/CVF international conference on computer vision workshops (ICCVW)*, (Montreal, BC), 1905–1914. doi: 10.1109/ICCVW54120.2021.00217

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., et al. (2018). ESRGAN: Enhanced super-resolution generative adversarial networks. *arXiv* [Preprint] arXiv:1809.00219.

Ward, M., Tulloch, A. I. T., Radford, J. Q., Williams, B. A., Reside, A. E., Macdonald, S. L., et al. (2020). Impact of 2019–2020 mega-fires on Australian fauna habitat. *Nat. Ecol. Evol.* 4, 1321–1326. doi: 10.1038/s41559-020-1251-1

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

Xue, Z., Lin, H., and Wang, F. (2022). A small target forest fire detection model based on YOLOv5 improvement. *Forests* 13:1332. doi: 10.3390/f13081332

Yuan, F., Zhang, L., Wan, B., Xia, X., and Shi, J. (2019). Convolutional neural networks based on multi-scale additive merging layers for visual smoke recognition. *Mach. Vision Appl.* 30, 345–358. doi: 10.1007/s00138-018-0990-3

Zhan, J., Hu, Y., Zhou, G., Wang, Y., Cai, W., and Li, L. (2022). A high-precision forest fire smoke detection ap-proach based on ARGNet. *Comput. Electr. Agric.* 196:106874. doi: 10.1016/j.compag.2022.106874

Zhang, R., Zhang, W., Liu, Y., Li, P., and Zhao, J. (2022). An efficient deep neural network with color-weighted loss for fire detection. *Multimed. Tools Appl.* 81, 39695–39713. doi: 10.1007/s11042-022-12861-9

Zhang, S., Wen, L., Bian, X., Lei, Z., and Li, S. Z. (2018). "Single-shot refinement neural network for object detection," in *Proceedings of the 2018 IEEE conference on computer vision and pattern recognition (CVPR)*, (Salt Lake City, UT), 4203–4212. doi: 10.1109/CVPR.2018.00442

Zheng, X., Chen, F., Lou, L., Cheng, P., and Huang, Y. (2022). Real-time detection of full-scale forest fire smoke based on deep convolution neural network. *Remote Sens.* 14:536. doi: 10.3390/rs14030536