Mathematical Biosciences and Engineering

MBE, 19(12): 13582–13606.
DOI: 10.3934/mbe.2022634
Received: 20 June 2022
Revised: 27 August 2022
Accepted: 31 August 2022
Published: 15 September 2022

http://www.aimspress.com/journal/MBE

Research article

Combination of UAV and Raspberry Pi 4B: Airspace detection of red imported fire ant nests using an improved YOLOv4 model

Xiaotang Liu[1], Zheng Xing[1,2], Huanai Liu[3], Hongxing Peng[2,4,5,*], Huiming Xu[2], Jingqi Yuan[2] and Zhiyu Gou[2]

[1] College of Materials and Energy, South China Agricultural University, Guangzhou 510642, China
[2] College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China
[3] School of Chemistry and Chemical Engineering, South China Technology of University, Guangzhou 510641, China
[4] Key Laboratory of Smart Agricultural Technology in Tropical South China, Ministry of Agriculture and Rural Affairs, Guangzhou 510642, China
[5] Guangzhou Key Laboratory of Intelligent Agriculture, Guangzhou 510642, China

* **Correspondence:** Email: xyphx@scau.edu.cn.

Abstract: Red imported fire ants (RIFA) are an alien invasive pest that can cause serious ecosystem damage. Timely detection, location and elimination of RIFA nests can further control the spread of RIFA. In order to accurately locate the RIFA nests, this paper proposes an improved deep learning method of YOLOv4. The specific methods were as follows: 1) We improved GhostBottleNeck (GBN) and replaced the original CSP block of YOLOv4, so as to compress the network scale and reduce the consumption of computing resources. 2) An Efficient Channel Attention (ECA) mechanism was introduced into GBN to enhance the feature extraction ability of the model. 3) We used Equalized Focal Loss to reduce the loss value of background noise. 4) We increased and improved the upsampling operation of YOLOv4 to enhance the understanding of multi-layer semantic features to the whole network. 5) CutMix was added in the model training process to improve the model's ability to identify occluded objects. The parameters of improved YOLOv4 were greatly reduced, and the abilities to locate and extract edge features were enhanced. Meanwhile, we used an unmanned aerial vehicle (UAV) to collect images of RIFA nests with different heights and scenes, and we made the RIFA nests (RIFAN) airspace dataset. On the RIFAN dataset, through qualitative analysis of the evaluation indicators, mean average precision (MAP) of the improved YOLOv4 model

reaches 99.26%, which is 5.9% higher than the original algorithm. Moreover, compared with Faster R-CNN, SSD and other algorithms, improved YOLOv4 has achieved excellent results. Finally, we transplanted the model to the embedded device Raspberry Pi 4B and assembled it on the UAV, using the model's lightweight and high-efficiency features to achieve flexible and fast flight detection of RIFA nests.

Keywords: RIFA nests; deep learning; improved YOLOv4; data augmentation; UAV

1. Introduction

Red imported fire ants **(**RIFA) are a socially harmful insect originating from the Paraguay River Basin [1]. They have spread to North America, Asia and other places through natural and human means, posing a huge threat to the safety of human life and property, as well as agriculture and forestry [2]. RIFA have been listed as one of the most destructive invasive species by the International Union for Conservation of Nature (IUCN). Prevention and control measures should be taken in time to reduce the loss caused by RIFA. Directly destroying RIFA nests can inhibit the speed of RIFA spread from the source. At present, researchers mainly use the manual observation method and the spectrometer detection method to locate and detect the RIFA nests. Wu et al. [2] analyzed the spectral information of the RIFA nests using a hyperspectrometer and used the differential method and the logarithmic method to process the reflection band. The results were verified by Euclidean distance, which could quickly distinguish the RIFA nests. However, these methods have the disadvantages of being time-consuming and having high cost and poor flexibility. An efficient and accurate detection method of RIFA nests is urgently required to reduce the difficulty of field exploration and improve operational efficiency.

In recent years, the application of machine vision and deep learning in agriculture has continued to deepen. There are two main object detection methods based on deep learning: One is an algorithm based on region generation, represented by R-CNN [3], Fast R-CNN [4], Faster R-CNN [5], Mask-RCNN [6]; and the other is based on an object regression algorithm, represented by You Only Look Once (YOLO) [7] and Single Shot MultiBox Detector (SSD) [8]. Roy et al. [9,10] mainly integrated DenseNet on YOLOv4 to optimize feature transfer, combined with modified PANet to retain fine-grained local information to achieve efficient detection of mango growth stages and plant diseases. Lawal [11] proposed an improved YOLO-Tomato model which mainly improved YOLOv3 by integrating dense architectures and spatial pyramid pooling. Experiments showed that this method can achieve higher accuracy and detection speed. Wu et al. [12] used the channel pruning YOLOv4 algorithm to detect apple blossoms in real time and constructed the YOLOv4 model under the CSPDarkNet53 framework. To simplify the apple blossom detection model and ensure the efficiency of the model, the channel pruning algorithm is used to prune the model, and the average accuracy of the final detection is 97.31%. The above methods use densely connected networks or model pruning to reduce the amount of network parameters, but the backbone network itself has not changed, and there is still more redundancy. In addition, the network feature extraction ability has not been significantly improved and is subject to the limitation of algorithm classification ability. Once there are occlusion problems, obvious changes in illumination and high similarity, the recognition accuracy will decrease, and the detection task cannot be efficiently completed in the

complex environment.

At present, the lightweight problem of a Convolutional Neural Network (CNN) has increasingly become the focus of research. The lightweight operation optimizes the effective feature layer and computational complexity of the traditional neural network, maintains reasonable accuracy while improving the running speed and provides conditions for porting to embedded systems [13]. Commonly used lightweight networks include MobileNet series [14], ShuffleNet series [15], EfficientNet [16], etc. These networks mainly rely on depthwise and pointwise operations of depth separable convolution to reduce the amount of network parameters and operations, or they use scaling factors to balance depth, width and resolution to compress the overall network framework. Zhang et al. [17] used MobileNetV3 as the feature extraction layer of YOLOv4. When the overlap was 0.50 on the cherry tomatoes test set, the accuracy was improved by 6.55% compared with YOLOv4, and the model was about 1/5 of that of YOLOv4. Mesut et al. [18] used the ShuffleNet model to create cloud datasets and extract features. Finally, the overall accuracy rates on the two cloud datasets were 98.56% and 100%, respectively, which has strong practical significance for climate monitoring, aerial flight and other fields. However, the above methods do not effectively solve the appearance of redundant feature maps, and they still generate redundant computing costs, which makes the occupation of invalid resources in the process of training and detection higher.

In addition, the light weight of the model also lays the foundation for intelligent field operations. Field operations often require sophisticated equipment, high flexibility and low cost, which makes a large number of embedded equipment combined with artificial intelligence algorithms. Wang et al. [19] proposed the lightweight YOLO-LiteDense and Yolo-DenseNano models to address the challenges of small objects in aerial images, which were implemented by adding dense connections and channel pruning on YOLOv3. The results showed that the model achieved higher accuracy and faster detection effect under the UAV platform. In order to solve the detection problem of metal corrosion by Micro Aerial Vehicles (MAVs), Yu et al. [20] combined various techniques such as CBAM attention mechanism to improve the detection accuracy of YOLOv3-tiny, and they improved SPP to fuse local features to further improve the detection accuracy of the model. Finally, the average accuracy of the improved AMCD model reached 84.96%, and it achieved faster detection performance. Cheng et al. [21] proposed an improved YOLOv4-tiny object detection algorithm for unmanned aerial vehicle (UAV) obstacle avoidance, using the depth separable convolution feature of MobileNet to reduce the amount of network parameters and using the least squares method as the position calculation of the three-dimensional coordinates. The final measured mAP of the model is 69%, and it is equipped with a UAV communication module for visual anti-collision testing. Meanwhile, the results confirm the effectiveness of the method. Victor et al. [22] used a convolutional neural network with depthwise separable convolutions to detect tomato leaf disease, achieved high accuracy and ported the model to a Raspberry Pi 4 microcomputer for future field deployment and detection.

In order to solve the above problems and realize the combination of algorithms and embedded devices, this paper proposes a RIFA nests detection model based on deep learning, and it uses the combination of UAV and Raspberry Pi 4B to further expand the application scenarios of the model. The main contributions of this paper are as follows:

1) In order to solve the problem of efficient and accurate identification of RIFA nests with less feature information, occlusion and adhesion and high similarity in complex wild scenes, this paper proposes an improved lightweight model of YOLOv4.

2) The backbone of YOLOv4 adopts the improved GhostBottleNeck to reduce the redundancy of generating feature maps and adds an Efficient Channel Attention mechanism to GBN for obtaining more effective feature information.

3) The Equalized Focal Loss in the head network was increased to reduce the weight of disturbing negative samples.

4) We adjusted the input size of improved YOLOv4 and added an upsampling module to the Neck network to add more effective feature information.

5) We added the CutMix data augmentation algorithm in the model training stage, and we evaluated and analyzed the model's detection effect on the RIFA ant nest in the RIFAN airspace dataset through experiments.

6) We used a Raspberry Pi 4B combined with the UAV for flying tests.

The rest of this paper is structured as follows: Section 2 describes the dataset, data augmentation methods and methods to improve the model. The experimental results, discussion of the model and the details of the hardware application are presented in Section 3. The conclusion is given in Section 4.

2. Materials and methods

2.1. Datasets

The experimental data in this paper were collected at the RIFA Research Base in the Taishan District of South China Agricultural University. The mature ant nest is 10–30 cm high and 30–50 cm in diameter. The shooting times were 9 September 2021 and 14 April 2022, both sunny days without clouds. In cooperation with RIFA research experts to identify, the modified DJI Phantom 4 UAV was used to shoot at different distances, occlusion degrees and lighting conditions. The modified UAV parts are shown in Figure 1. Among them, (a) in Figure 1 is the modified UAV, including 5V, 3A mobile power, detachable node and Raspberry Pi 4B, while (b) in Figure 1 is the Raspberry Pi 4B used in this paper, which is an easy-to-transplant embedded device, like a microcomputer. In order to maintain the stability of the landing, we set the bracket so that the four limbs touch the ground. The length of one bracket is 30.7 cm, the height is 5.8 cm, and the distance between the brackets on both sides is 19.7 cm. The four corners of the Raspberry Pi 4B are fixed with screws and connected to the chassis on the bracket. Meanwhile, in order to facilitate the installation of the device, we designed a detachable node, as shown in Figure 1(c).

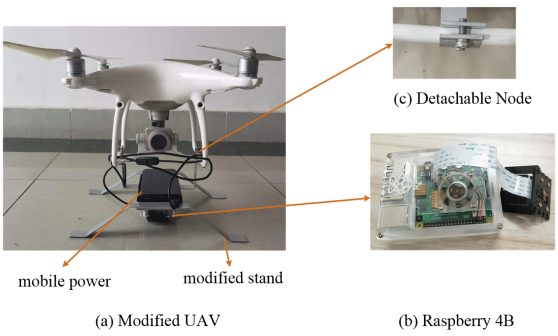

Figure 1. Modified UAV parts diagram: (a) Modified UAV, (b) Raspberry 4B, (c) Detachable Node.

The UAV flying was set to three heights: 1–3 m, 3–5 m and 5–7 m, and the resolution is 4096 × 2160 pixels. After screening and sorting, a total of 3202 images of RIFA nests were obtained, and some of the collected images are shown in Figure 2. Table 1 shows the distribution of the number of datasets. We divide the dataset into training set, validation set and test set according to the ratio of 8:1:1.

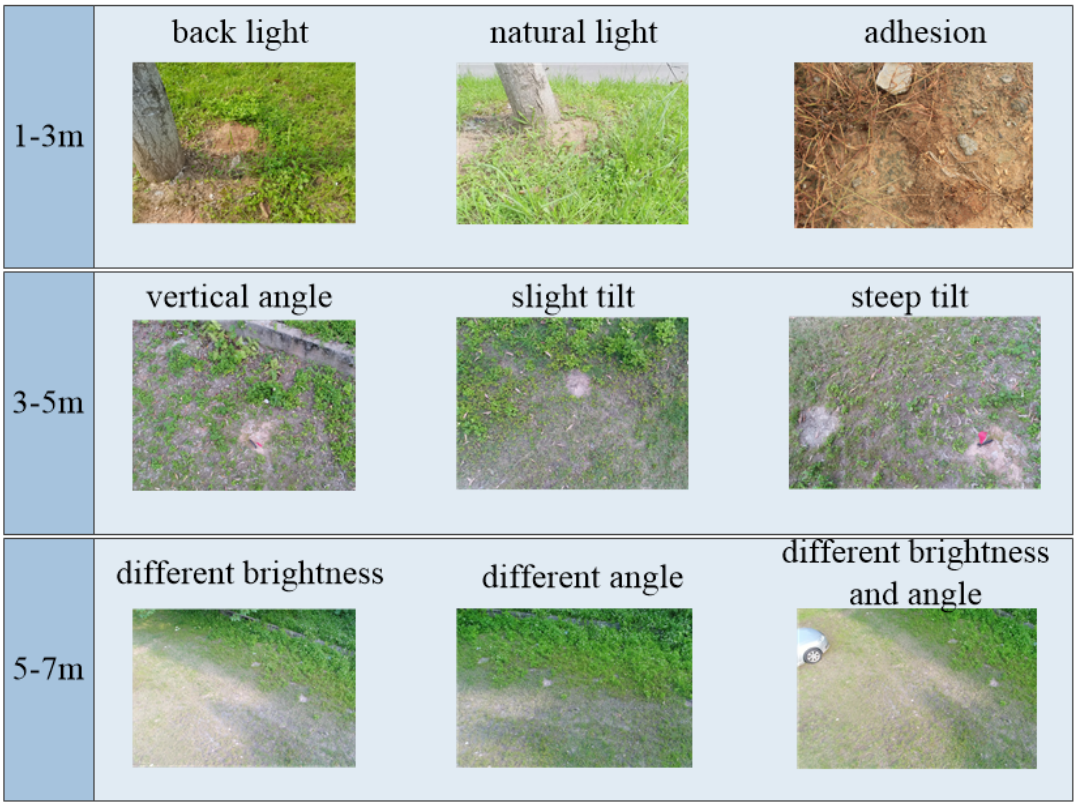

Figure 2. Classification of datasets.

Table 1. Number of RIFAN in the dataset.

Dataset	1–3 m	3–5 m	5–7 m	Total
Training set	1867	321	373	2561
Validation set	233	40	46	319
Test set	234	41	47	322
Total	2334	402	466	3202

2.2. Data augmentation

Due to the large number of negative samples collected in the field, there is a category imbalance problem that hinders the detection of ant nests. In order to solve the above problems and expand the dataset, this paper adopts the method of data augmentation, through the use of random rotation, translation, segmentation, median filtering, Gaussian noise, dilation and erosion, etc., to reduce the phenomenon of model over-fitting and improve the generalization ability. In addition, we also use the CutMix to randomly fill the cropped area of each image with pixel values to improve the

diversity of data around the object [23].

At the same time, we consider that some data augmentation may lead to changes in image size, so this paper uses the three-scale input method and sets the input size to 32 m × 32 m (m = 10, 15, 19), corresponding to 320 × 320, 480 × 480 and 608 × 608, respectively. It can make the feature information extracted by the network in different sizes more suitable. In practical applications, the three-scale input method can also solve the problem of the difference in resolutions of RIFA nest images obtained by different devices [24], and the effect is shown in Figure 3. Table 2 shows the changes of the data before and after data augmentation, and finally we obtain the RIFAN dataset containing 12,808 RIFA nests.

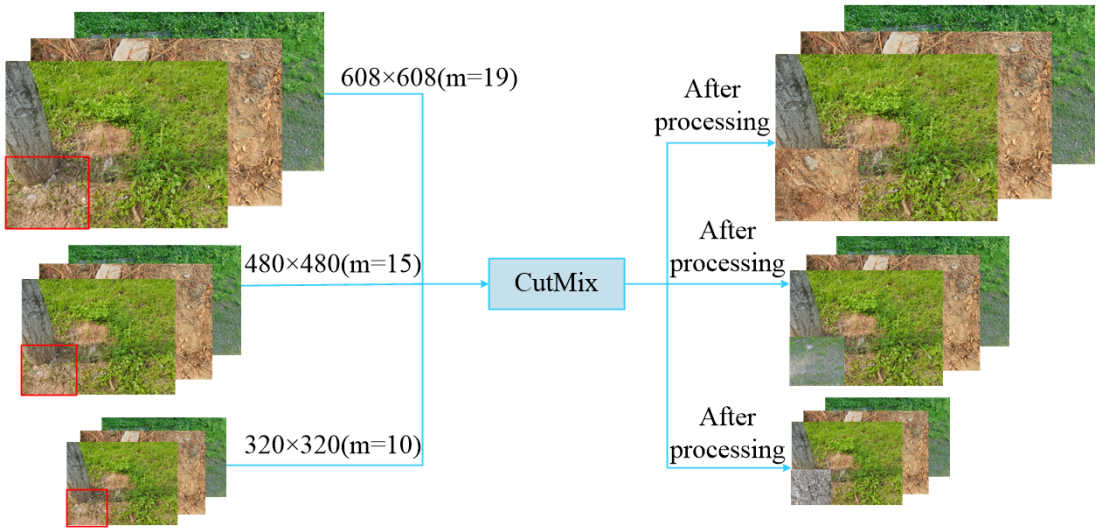

Figure 3. Changes of three-scale input under CutMix.

Table 2. Quantitative changes in the dataset before and after data augmentation.

Dataset	Without data augmentation	With data augmentation
1–3 m	2334	9336
3–5 m	402	1608
5–7 m	466	1864
Total	3202	12,808

2.3. YOLOv4 algorithm

Compared with other one-stage and two-stage algorithms [25], YOLOv4 has better inference speed and detection performance, and it can directly classify, predict and generate bounding boxes. The YOLOv4 structure diagram is shown in Figure 4. The input of YOLOv4 is mainly composed of convolution, batch normalization (BN) and Mish algorithm [26]. The 416 × 416 image is input into the network for preprocessing, and the image is scaled and normalized, which can improve the generalization ability of the network. The backbone extraction layer is composed of the CSPDarkNet53 network, including CSP1, CSP2, CSP8, CSP8 and CSP4, a total of 5 Cross Stage Partial modules (CSP). The number of each CSP corresponds to the corresponding number of residual units. The residual unit can build a deep network and form a CSP module with the Conv_BN_Mish (CBM) module through the Concat operation. The structures of the CSP module and the residual unit are shown in Figure 5. X in CSPX represents the number of residual units, each

CSPX includes five Conv_BN_Mish modules, and the size of the convolution kernel is 3 × 3. The residual unit consists of two Conv_BN_Mish sub-modules, which uses the add operation to add tensors without expanding the dimension.

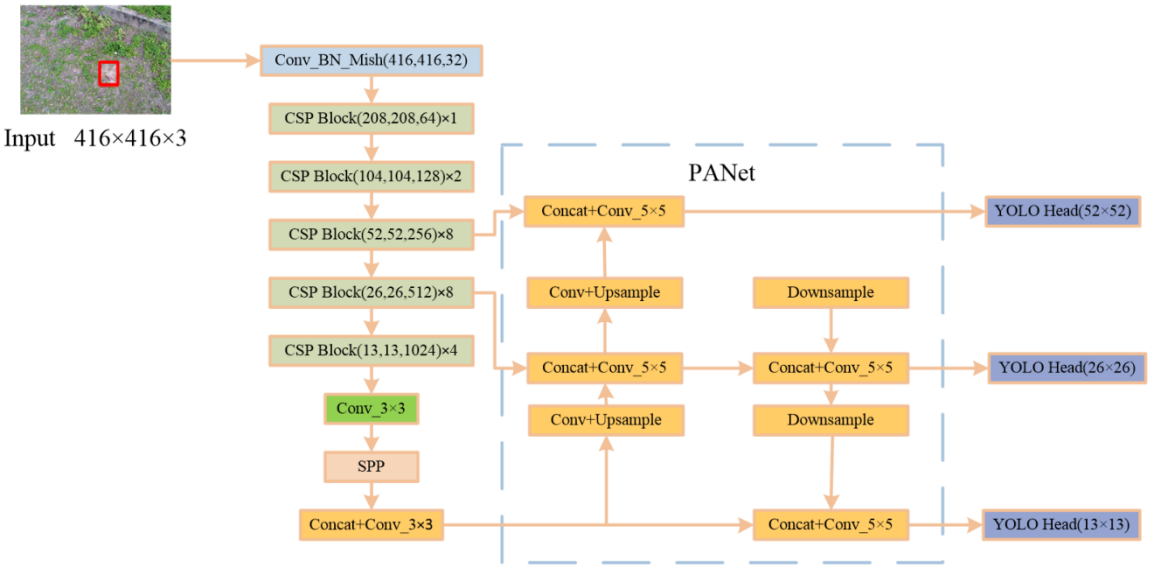

Figure 4. YOLOv4 structure diagram.

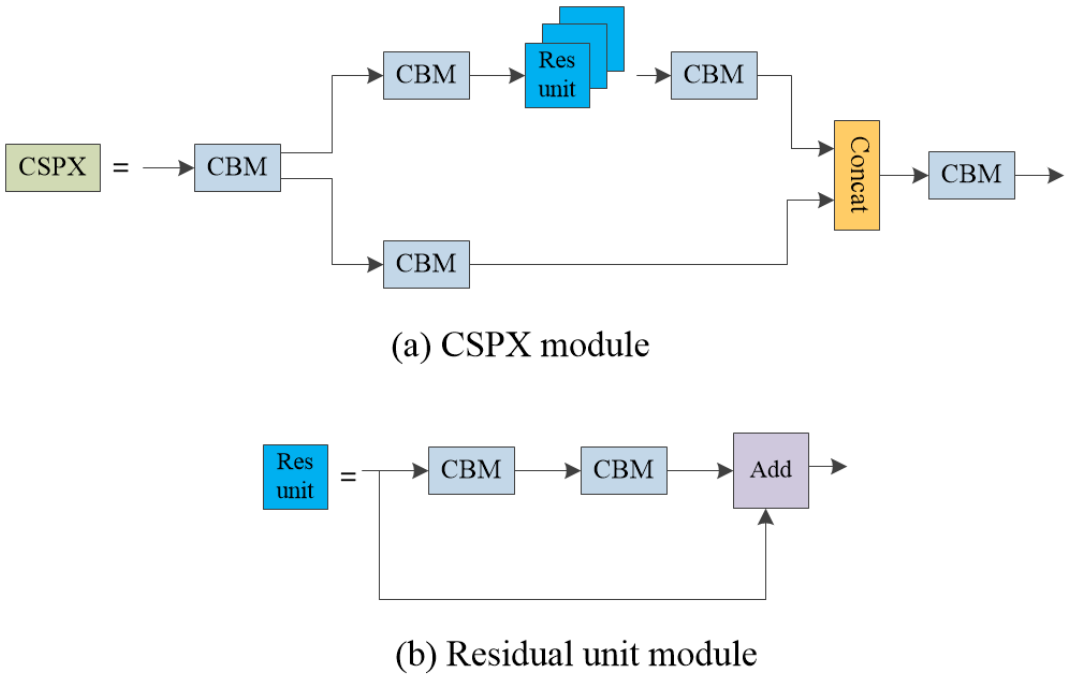

Figure 5. Structures of the CSP module and the residual unit: (a) CSPX module, (b) Residual unit module.

After the feature extraction of the 416 × 416 size image, the feature maps of 52 × 52 and 26 × 26 are input to the Path Aggregation Network (PANet). PANet can perform bottom-up and top-

down bidirectional fusion by using up and down sampling. Through the 5 × 5 convolution and the Concat operation of CSP module, the spatial information of features is preserved, and the propagation path between the convolution layers is shortened to enhance the network representation ability.

The YOLOv4 network also performs spatial pyramid pooling (SPP) processing on the 13 × 13 feature layer. The SPP structure is shown in Figure 6. Let k_i (i = 2, 3, 4) be the max pooling operation, and let y_i be the extracted image feature. Then, the relationship between the two is expressed by Eq (1).

$$y_i = \begin{cases} x, i = 1 \\ k_i \cdot x, i \in \{2,3,4\} \end{cases} \tag{1}$$

The image features processed by the four max pooling methods are combined by Concat and output as $y = \sum_{i=1}^{4} y_i$.

Compared with the traditional k × k max pooling method, SPP can realize the parallel operation of 1 × 1, 5 × 5, 9 × 9 and 13 × 13 max pooling, process the context information in layers and improve the scale range received by the feature layer, enhancing the robustness of the network to image scales.

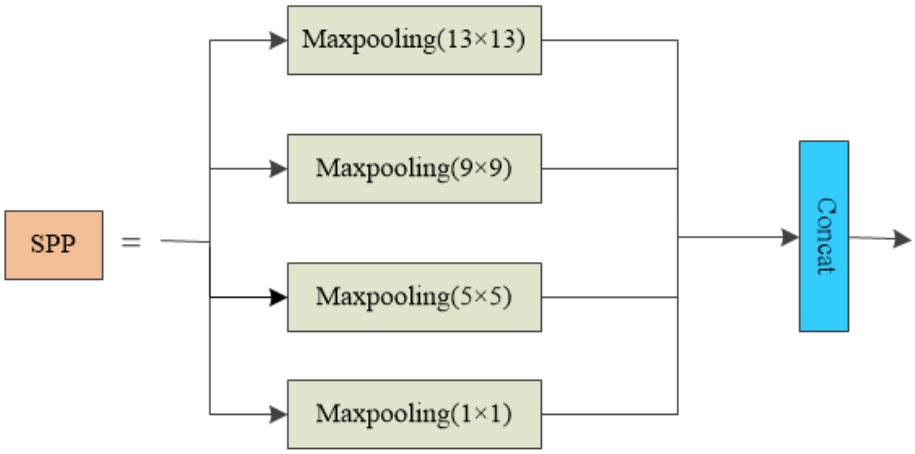

Figure 6. SPP structure.

After the feature information is processed by the backbone network and PANet, three feature maps are finally obtained. In order to comprehensively consider the length, width, height and position information of the bounding box, YOLOv4 uses the CIOU Loss function for anchor box regression, which improves the accuracy and speed of the prediction box regression. The formulas for calculating CIOU are as follows:

$$CIOU = IOU - R_{CIOU} \tag{2}$$

$$R_{CIOU} = \frac{\rho^2 \cdot (b, b^{gt})}{c^2} + \alpha v \tag{3}$$

$$\alpha = \frac{v}{1 - IOU + v} \tag{4}$$

$$v = \frac{4}{\pi^2} \cdot (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h})^2 \tag{5}$$

where R_{CIOU} is the regularization of CIOU, $\rho^2 \cdot (b, b^{gt})$ is the Euclidean distance between the center point of the predicted frame and the real frame in the image, c is the diagonal distance of the smallest closed area that includes both the predicted frame and the real frame, α is the weight function, w and h represent the width and height of the effective box, respectively, and v is the aggregate similarity of width and height. Combined with the formulas above, the complete formula for CIOU Loss regression can be obtained.

$$Loss_{CIOU} = 1 - IOU + \frac{\rho^2 \cdot (b, b^{gt})}{c^2} + \frac{v}{1 - IOU + v} \cdot \frac{4}{\pi^2} \cdot (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h})^2 \tag{6}$$

It can be seen from Eq (6) that the CIOU Loss function increases the aspect ratio information, so that the matching degree between the predicted frame and the real frame increases.

2.4. Proposed YOLOv4 algorithm

2.4.1. Improved GhostBottleNeck

The residual unit of the CSP block in YOLOv4 can reduce the amount of parameters, but it can still generate more computational costs. In this paper, we replace the CSP block with GBN, which can greatly reduce the computational complexity of the network and improve the model's running speed. The original GBN structure is shown in (a) of Figure 8. The Ghost module is the key of GBN, which uses a convolutional layer with few output feature maps and a simple linear transformation to replace the convolutional layer [27]. It not only reduces the redundancy of the feature map but also makes the network lightweight. The Ghost module structure diagram is shown in Figure 7. The operation of a is to use the feature map generated by the simple convolution of the image as the final output feature map, and b represents further linear transformation of the feature map with depthwise convolution to obtain more feature maps.

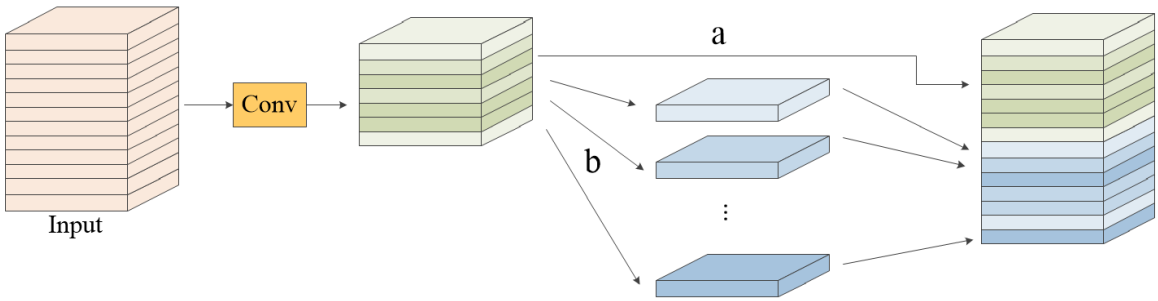

Figure 7. Ghost module.

Two strides of GBN are composed of two Ghost modules, the first Ghost module is processed by BN and ReLU, and the number of channels is increased. The second Ghost module is only processed by BN, and the number of channels is reduced. When stride is equal to 2, the depth convolution is added, which reduces the amount of parameters and improves the recognition accuracy.

If the input feature map size of the t layer is m × m × 3, and the output is n × n × C, the convolution calculation amount for the convolution kernel size of k × k is n × n × C × k × k × 3, while the calculation amount of the Ghost module is

$$n \times n \times \frac{C}{t} \times k \times k \times 3 + (t-1) \times n \times n \times \frac{C}{t} \times k \times k \tag{7}$$

It can be seen from Eq (7) that the Ghost module changes the multiplication operation of the original convolution to a simple convolution plus a depthwise convolution operation, which reduces the computational complexity. In addition, compared with similar MobileNet series lightweight networks or similar to deeper ResNet series networks, GBN reduces the generation of a large number of similar feature maps due to its unique Ghost module characteristics, which can fundamentally solve the redundancy problem of training feature maps. In addition, the presence of more background noise in the captured RIFAN dataset increases the burden of training, and generating a model with more parameters is not conducive to the application of the embedded side. However, using GBN with the Ghost module can effectively solve the above problems, which is the main reason why we use GBN.

We have studied GBN in depth and found that ReLU processing corresponding to the Ghost module has limitations in each stride. When the input pixel matrix is close to 0, there will be a phenomenon that back-propagation cannot be performed, resulting in the model learning surface not completely covering the features in the image, which is even worse for datasets with many negative samples. Therefore, we change each ReLU in GBN to LeakyReLU, and the formula is shown in Eq (8). It assigns a very small component of x to the negative direction, which can achieve fast back-propagation to the input of 0 or negative data.

$$Leaky\,\mathrm{Re}\,LU = \begin{cases} x, & x > 0 \\ 0.01x, & x \le 0 \end{cases} \tag{8}$$

Another reason why we choose LeakyReLU is that it maintains a low gradient and linear

propagation in the negative direction, which can promote faster convergence of the model [28], and it consumes less computing resources than activation functions such as PReLU and ELU. This provides more sufficient conditions for us to transplant the model to embedded devices such as the Raspberry Pi 4B.

It is worth noting that due to the addition of LeakyReLU, both positive and negative activation units are output through the Ghost module, which does not guarantee that the activation units can output effectively, and there may be omissions. Inspired by the fine-tuning operation of transfer learning, we consider adding the fully connected layer to GBN to capture all inputs, but its large number of parameters and serious over-fitting problems make us turn our attention to the global average pooling (GAP) layer. It can replace the fully connected layer and has a small amount of parameters. At the same time, it has a global receptive field, is sensitive to the spatial location information of the feature map and can directly regularize the entire network.

Due to the excellent characteristics of GAP, we also introduce the Efficient Channel Attention (ECA) mechanism with GAP in GBN [29] and transplant it to the front of each Ghost module in GBN. When the feature map goes through ECA, GAP is used to change the size of h × w × C to 1 × 1 × C. Afterwards, in order to facilitate cross-validation to select one-dimensional convolution kernel size k × k, an adaptive selection method of the k value is developed. The formula of k is

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

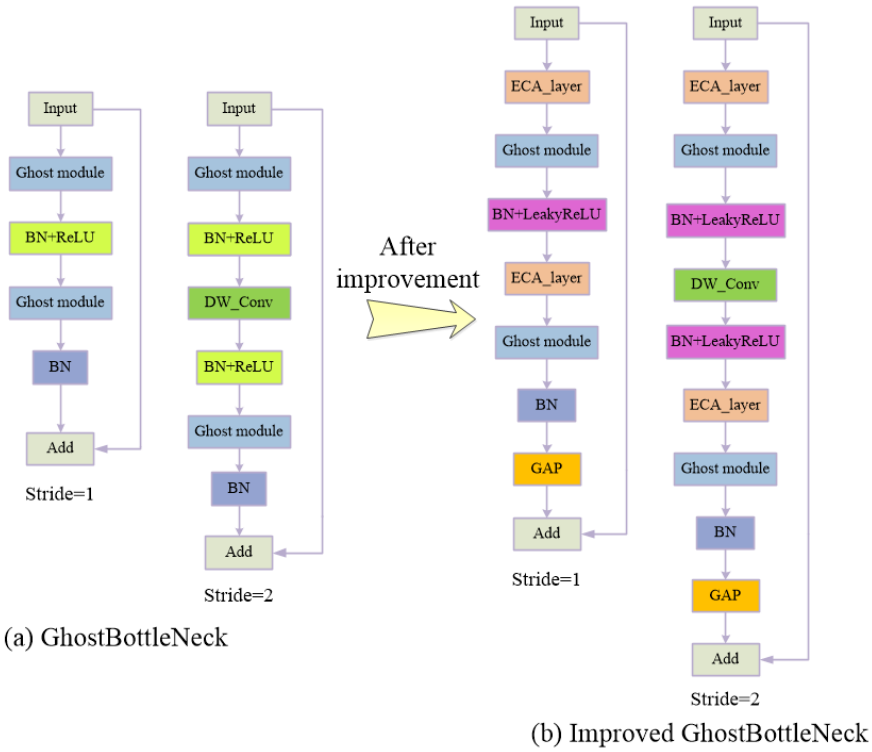

Figure 8. GhostBottleNeck before and after improvement: (a) GhostBottleNeck, (b) Improved GhostBottleNeck.

In Eq (9), k is proportional to the channel, the $|\ |_{odd}$ operator means to take the odd number closest to the number in the symbol, C is the number of channels, γ = 2, b = 1. Compared with Squeeze and Excitation (SE), Convolutional Block Attention Module (CBAM), Coordinate Attention (CA) and other methods, ECA utilizes an unreduced cross-channel interaction strategy, determines the coverage of channel interaction through adaptive one-dimensional convolution and reduces redundant calculations and network complexity while maintaining performance. The improved GBN is shown in Figure 8(b).

2.4.2. Improved head network

YOLOv4 will generate a large number of redundant candidate boxes during the detection process, which will cause the network to deviate from the positive sample optimization direction, resulting in poor recognition accuracy. In this case, there is a lack of a loss function to adjust the imbalance of positive and negative sample categories. Although Focal Loss can solve the class imbalance problem, it uses the same modulation factor for all classes and cannot fully adapt to the entire data sample. In order to solve this problem, this paper uses the Equalized Focal Loss (EFL) function and adds it to the head network of YOLOv4. Using its dynamic class modulation factor, combined with CIOU Loss to regress the prediction box, the weight of negative samples can be reduced [30]. EFL contains two kinds of adjustment factors. the first factor is the Focusing Factor, which is the adjustment parameter of difficult and easy sample aggregation, and the j-th category is represented as γ^j, as shown in Eq (10). It is split into two parameters, where γ_b is class-independent and γ_v^j is class-dependent. When $\gamma^j > 0$, the proportion of the loss of simple samples is reduced, and the entire network will pay more attention to the difficult-to-classify samples, so that the computational cost can be effectively allocated. The second factor is the Weighting Factor, which can balance the loss contributions of different categories. Here, we set the j-th category as $\frac{\gamma_b + \gamma_v^j}{\gamma_b}$, and the final representation of EFL is shown in Eq (11).

$$\gamma^j = \gamma_b + \gamma_v^j \tag{10}$$

$$EFL(p_t) = -\sum_{j=1}^{C} \alpha_t \left(\frac{\gamma_b + \gamma_v^j}{\gamma_b} \right) \cdot (1 - p_t)^{\gamma_b + \gamma_v^j} \log(p_t) \tag{11}$$

where p_t is the probability of sample prediction, and α_t is a factor that balances the uneven proportion of positive and negative samples. Since the negative samples are easy to divide, this paper reduces the proportion of positive samples and sets α_t to 0.25 to increase the penalty value when the prediction is wrong.

2.4.3. Improved YOLOv4 network

In this paper, we combine the improved GBN and head network and finally design the improved YOLOv4 network, as shown in Figure 9. The input uses a three-dimensional input method, set to 32 m × 32 m × 3 pixels, where m is 10, 15, and 19, covering three common sizes. We set the improved GBN as ECA_GBN and change the feature map size of each ECA_GBN to prevent network

redundancy. CBL represents a module composed of convolution, BN and LeakyReLU.

To enhance the feature extraction ability, we add upsampling after the CBL corresponding to the third and fourth ECA_GBN, and the upsampling process is shown in Figure 9. The upsampling method in this paper is to copy and fill the upper-level pixel information into the feature map of the object area. This copying operation is simple and convenient, and it reduces the consumption of computing resources. Meanwhile, upsampling can make the resolution of the feature map higher, obtain more favorable feature information and increase the understanding of the multi-layer semantic features of the Neck network. Finally, in order to optimize the head network, we combine the Equalized Focal Loss function and CIOU Loss to make the results closer to the ground truth.

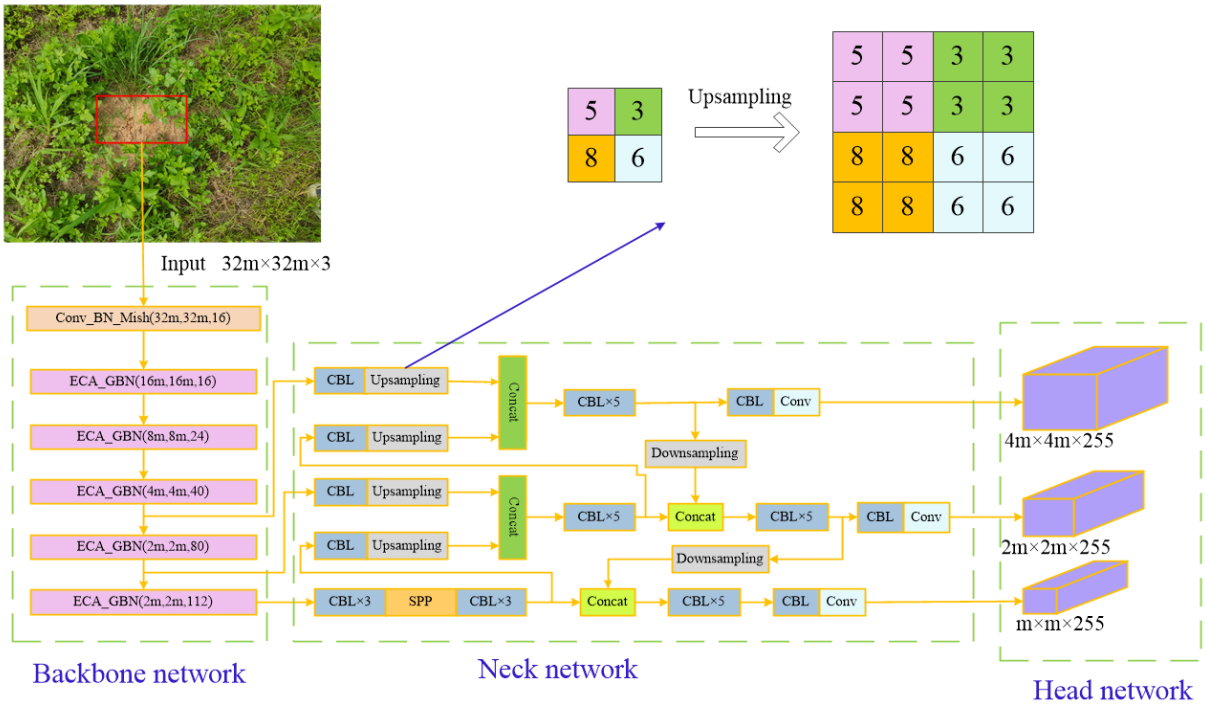

Figure 9. Proposed YOLOv4 structure diagram.

3. Experimental results and discussion

3.1. Experimental setting

This section mainly compares and analyzes improved YOLOv4 with various networks. For the input size of 32 m × 32 m × 3 (m = 10, 15, 19), m is randomly selected every 10 epochs. The software environment for model training and testing is Ubuntu 18.04.5, conda 4.10.1, python 3.7.10, pytorch 1.8.1, PyCharm2021.1.1. The hardware environment includes an Intel Core i9-10900K, 3.7 GHz × 20 processing frequency, 64 G running memory, NVIDIA GeForce 3090 graphics card and 24G video memory.

The hyperparameter settings in the training process are shown in Table 3. First, improved YOLOv4 is trained on the PASCAL VOC2007 dataset to obtain the pre-training weights of the improved algorithm. The transfer learning method is used to initialize the model with pre-training

weights, which can increase the training speed. The total number of epochs is set to 200, of which the first 50 epochs freeze the pre-training weights, batch size is set to 16, the initial learning rate is 0.001, and the weight decay factor is 0.0005. The last 150 epochs are training after thawing, batch size is 8, the initial learning rate is 0.0001, and the weight decay factor is 0.0005. The SGD (Stochastic gradient descent) optimization algorithm and label smoothing method are added to the training to prevent over-fitting.

Table 3. Hyperparameter value set.

Hyperparameter	Values
epochs	200
batch size	16- > 8
learning rate	0.001- > 0.0001
weight decay factor	0.0005
optimization algorithm	SGD

In order to objectively measure the effect of improved YOLOv4 in detecting RIFA nests, this paper uses six quality indicators: Precision, Recall, F1 score, Mean Average Precision (MAP), Parameter and detection speed. The formulas of Precision, Recall, F1 score and MAP are as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\% \tag{12}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% \tag{13}$$

$$\text{F1} = \frac{2\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\% \tag{14}$$

$$\text{mAP} = \int_0^1 (\text{Precision} \cdot \text{Recall})\ \text{dRecall} \times 100\% \tag{15}$$

where TP represents the number of correctly predicted positive samples, FP represents the number of incorrectly predicted negative samples as positive samples, FN represents the number of incorrectly predicted positive samples as negative samples, F1 represents the harmonic mean of Precision and Recall, MAP represents the average accuracy of detection, the detection speed is set to the number of frames read per second (frames/s), and Parameter represents the total number of variables added during the network learning process.

3.2. Training process analysis

The loss values of the training set and validation set and the loss value after adding label smoothing are shown in Figure 10. During the test, the loss value without label smoothing tends to oscillate, but it is already in the fitting state at epoch = 125. In order to reduce the over-fitting phenomenon, the label smoothing method is added, and noise is added through the soft one-hot internal algorithm to reduce the loss value, so that the network has local jump changes at epoch = 6 but then tends to be stable, and the model converges best at epoch = 200.

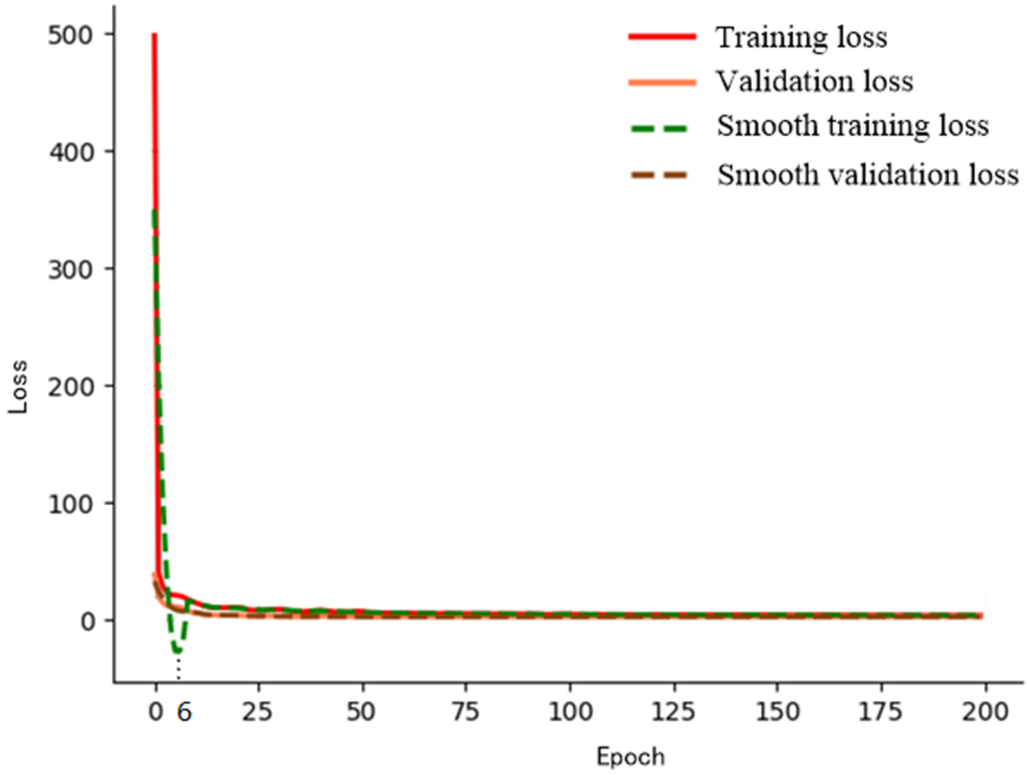

Figure 10. Loss curve change graph during training.

3.3. Performance comparison

3.3.1. Model comparison

First, in order to test the effectiveness of the improved algorithm, we compare the proposed method and YOLOv4 on the test set, and the results are shown in Table 4. In terms of detection accuracy, improved YOLOv4 has a precision of 97.82%, a recall of 97.44%, an F1 of 98% and a MAP of 99.26%, which are 6.01%, 6.13%, 6% and 5.9% higher than those of YOLOv4, respectively. The reason why these indicators perform well is that this paper uses the improved GBN to extract features, and the improved Neck network adjusts the range of the receptive field, adapts to multi-scale input, reduces the loss of position information in the feature map and uses the cross-channel interaction strategy of ECA module to enhance the extraction of key information from the object image and improve the accuracy of model detection. In terms of detection speed and parameters, the detection speed of the improved YOLOv4 model is 30 frames/s, and the parameters are 44.6 MB, which is 8 frames/s higher than YOLOv4 and reduced by 211.7 MB. This is due to the light weight of the Ghost module, the adaptive one-dimensional convolution kernel in the ECA channel and the use of Equalized Focal Loss to dynamically adjust the category imbalance and reduce the generation of redundant candidate boxes. In general, parameters are reduced, and the detection speed is improved.

Figures 11 and 12 show the change curves at different score thresholds of the four indicators and P-R curves before and after the improvement of YOLOv4. In this paper, the score threshold of

the detection box is set to 0.5, and the boxes less than 0.5 are filtered out. As can be seen from the figure, YOLOv4 is prone to small fluctuations when the score threshold is less than 0.5 and the change of early recall. This is rarely seen in the improved YOLOv4, because the original model is not robust to feature maps of different scales, and the extracted effective information is different and complex, which will interfere with the learning process of the model, resulting in the phenomenon of curve fluctuation. Improved YOLOv4 has added a multi-scale input method that can accurately locate the region of interest of the feature map and adapt to the transformation of multi-scale feature maps. As a result, its accuracy, recall, F1 and MAP training fluctuate less and basically maintain a stable state, and the model has strong robustness and generalization ability.

Table 4. Comparison of indicators before and after YOLOv4 improvement.

Models	Precision	F1	Recall	MAP	Parameters	FPS
YOLOv4	91.81%	92%	91.31%	93.36%	256.3 M	22
Proposed method	97.82%	98%	97.44%	99.26%	44.6 M	30

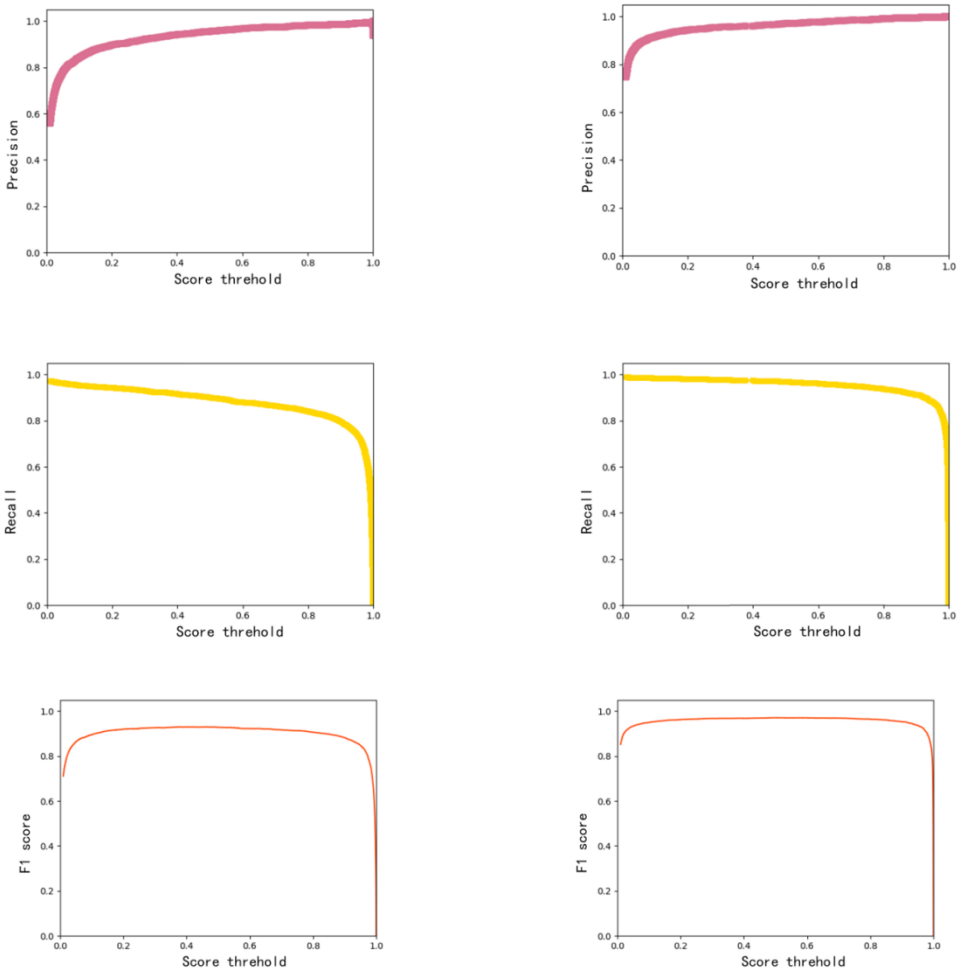

Figure 11. Changed curves of different score thresholds.

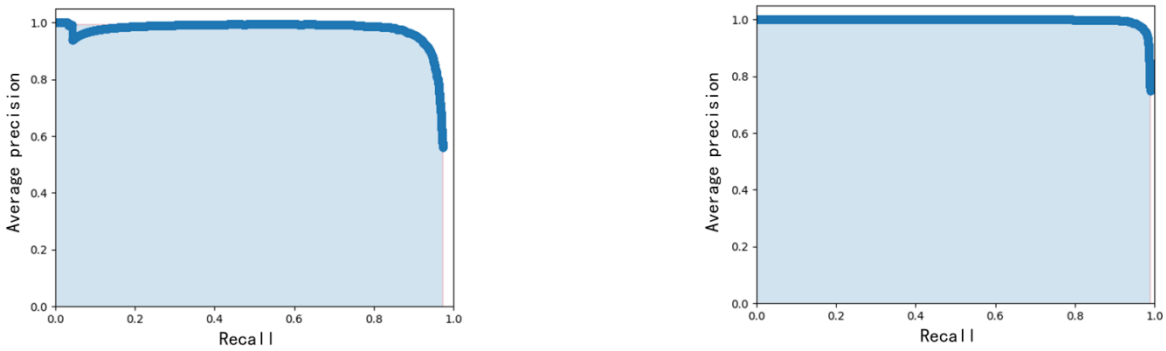

Figure 12. Precision-Recall curves.

The proposed model can improve the accuracy of predicting bounding boxes at different heights because of the more upsampling operations in the Neck network. In order to verify this performance of the model, this paper tests the IOU value of the model before and after the improvement, and the results are shown in Figure 13. IOU can reflect the coincidence ratio between the predicted frame and the real frame. The higher IOU is, the higher the accuracy of the model's prediction to the bounding box will be. It can be seen from Figure 13 that the improved model has a high IOU value for different heights, can obtain more semantic information at key points in different scenarios and performs better.

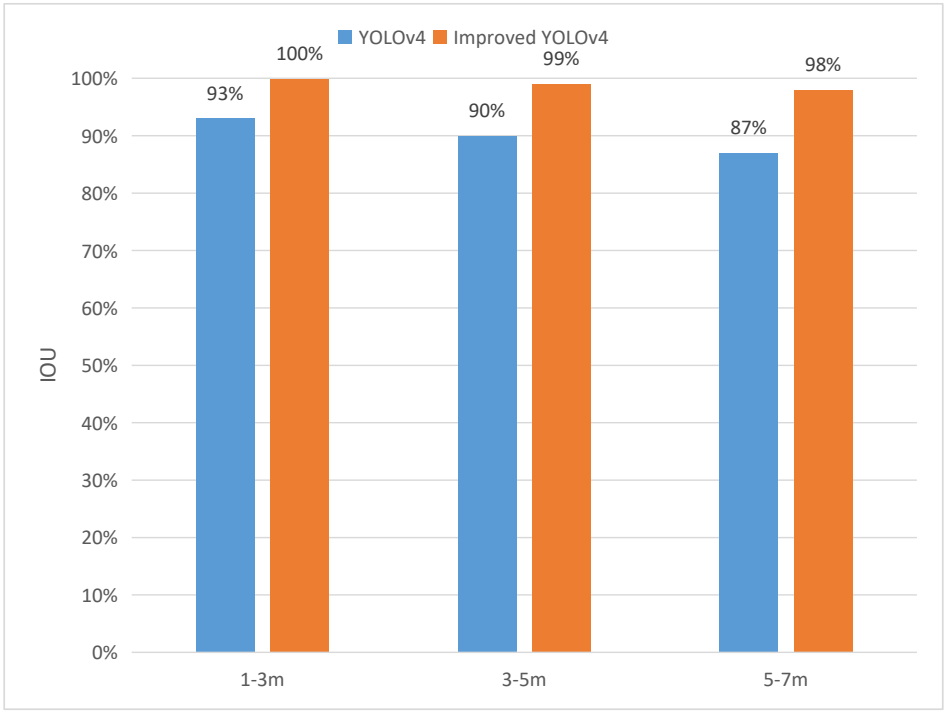

Figure 13. Column chart of changes in IOU.

In order to verify our improved GBN, we replace the YOLOv4 backbone network with different networks for experiments, including the commonly used dense layer network and

lightweight network. The control variable method is adopted to keep the same as the other methods, and the experimental results are shown in Table 5. From the table, we can clearly find that improved YOLOv4 has a small range of improvement in the five indicators compared with other networks, and the performance is ideal. On the one hand, for the unimproved GBN model, although its recall is 0.47% lower than that of MobileNetv1, and its MAP is 0.04% lower, its precision is 0.18% higher, the frame rate is increased by 1, and parameters are reduced by 9.3 M. These advantages are very important for the transplantation of embedded devices, the comprehensive performance is better, and the elaboration compared with other networks is similar. On the other hand, we can also find that, compared with the unimproved GBN model, the performance of the improved GBN has improved significantly. Especially, the parameter amount has only increased by 0.3 M, and the increased part accounts for a very small proportion of the overall network. Within the control range, it meets the needs of the mobile terminal.

Table 5. Comparison of different backbone networks under YOLOv4.

Models	Precision	F1	Recall	MAP	Parameters	FPS
YOLOv4 + MobileNetv1	97.1%	97%	97.79%	99.18%	53.6 M	27
YOLOv4 + ResNet50	96.02%	90%	84.46%	93.36%	133.6 M	26
YOLOv4 + MobileNetv2	96.34%	97%	97.9%	99.09%	48.8 M	28
YOLOv4 + CSPDarkNet53	95.64%	93%	90.09%	95.67%	256.5 M	24
YOLOv4 + GhostNet	97.28%	97%	97.32%	99.14%	44.3 M	28
YOLOv4 + VGG16	96.1%	88%	80.96%	91.36%	94.3 M	25
YOLOv4 + DenseNet121	95.73%	86%	78.56%	90.57%	65 M	26
Proposed method	97.82%	98%	97.44%	99.26%	44.6 M	30

YOLOv4 has the advantages of fast speed, high regression accuracy of object bounding boxes and low training threshold. Coupled with our improvements, YOLOv4 pays more attention to the object and adapts to multi-layer semantic features. We compare the improved model with some one-stage and two-stage algorithms and add pre-trained weights, and the results are shown in Table 6 and Figure 14. Since we have improved GBN from the perspective of reducing the number of feature maps and convolutions, the entire model has become lightweight, the number of model parameters is only 44.6 M, the four indicators detected are also very good, and the difference compared with other algorithms is significant. In addition, we also use the log average miss rate to test the performance of the model. From Figure 14, we can see that the false detection rate of the method proposed in this paper is 0.01, which is 0.12 lower than that of YOLOv4, the sample size of the false prediction is reduced, the difference between the false detection rate and other algorithms is more than 0.05, and the performance is better.

Table 6. Comparison results with popular algorithms.

Models	Precision	F1	Recall	MAP	Parameters
YOLOv4	91.81%	92%	91.31%	93.36%	256.3 M
SSD	96.15%	87%	79.71%	91.54%	95 M
YOLOv3	98.04%	86%	76.81%	90.85%	246.5 M
CenterNet	97.52%	92%	86.21%	95.97%	765.8 M
YOLOv5	95.47%	91%	87.53%	95.85%	84.6 M
Retiannet	93.59%	80%	70.24%	71.36%	145.7 M
Faster R-CNN	52.68%	67%	93.52%	89.75%	546.8 M
Proposed method	97.82%	98%	97.44%	99.26%	44.6 M

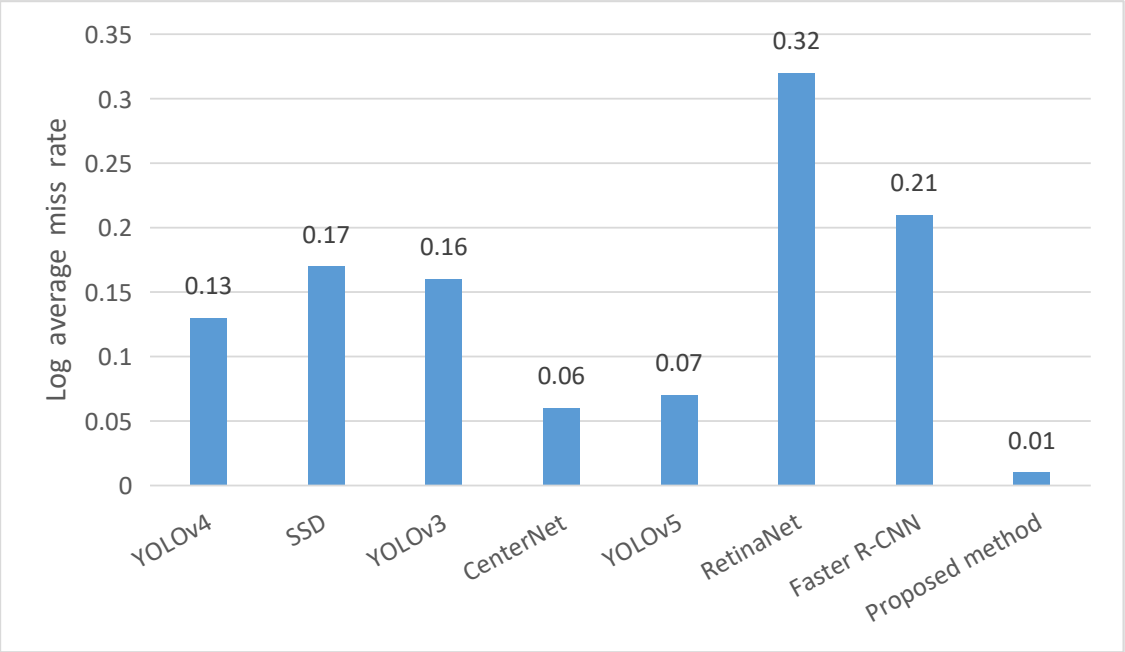

Figure 14. Column chart of log average miss rates.

Considering the complexity of the wild environment, we added the CutMix data enhancement method before model training and selected random clipping and replacement around the object to increase the robustness of the model to negative samples and improve the quality of the RIFAN dataset. To further verify the effectiveness of the method, we conduct relevant comparative experiments, as shown in Table 7. It can be noticed that after using the CutMix method, the performance of our proposed model has been slightly improved, with an average increase of 0.535% for the four indicators, which is critical for the success rate of detection.

Table 7. Comparison of network gain effects with CutMix.

Models	Precision	F1	Recall	MAP
With CutMix	97.82%	98%	97.44%	99.26%
Without CutMix	97.69%	97%	96.74%	98.95%

3.3.2. Ablation experiment

We verify the detection effect of the improved GBN in Table 5. In this section, we will further evaluate and analyze the two improved methods of ECA and Equalized Focal Loss. Therefore, we conduct related ablation experiments, and the results are shown in Table 8. We can see that after adding ECA, precision, F1 and MAP of YOLOv4 have increased, and parameters have only increased by 0.9 M. However, recall has decreased by 1.9%. This is due to class imbalance. The model incorrectly predicts positive samples as negative samples, and the number of such phenomena increases, resulting in lower recall. When the Equalized Focal Loss function is added alone, the precision of the model drops by 1.62%. This takes into account that the algorithm is not sensitive to the features of positive samples and has less attention, which leads to the phenomenon of more erroneously predicting the number of negative samples as positive samples. After using the method proposed in this paper, the above problems can be effectively solved, and two methods can synergize with each other and learn from each other's strengths.

Table 8. Comparative Results of ablation experiments.

Models	Precision	F1	Recall	MAP	Parameters
YOLOv4	91.81%	92%	91.31%	93.36%	256.3 M
YOLOv4 + ECA	92.57%	93%	89.41%	95.24%	257.2 M
YOLOv4 + Equalized Focal Loss	90.19%	92%	94.38%	94.74%	256.3 M
YOLOv4 + ECA + Equalized Focal Loss (Proposed method)	97.82%	98%	97.44%	99.26%	44.6 M

To further visualize the detection results, we test the four methods in Table 8 with heat maps, and the results are shown in Figure 15. Here, we can see that with the change of the model improvement method, the generation areas of heat maps have changed to varying degrees. Compared with the other three methods, YOLOv4 has a weaker detection ability, and the activation areas of heat maps are not obvious or even detected incorrectly, while the proposed method has a relatively complete coverage of the location of the RIFA nests during the detection process.

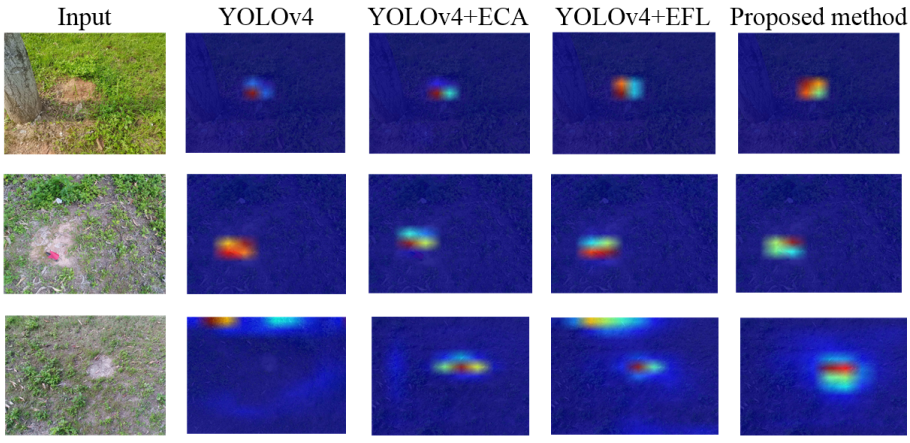

Figure 15. Comparison of heat maps effect.

3.3.3. Comparison of different attention mechanisms

ECA uses adaptive one-dimensional convolution to perform cross-channel interaction, which can make the model maintain a high-precision detection effect. To verify this improvement, we compare experiments with different attention mechanisms. The experimental results are shown in Table 9. We test these four sets of experiments on improved YOLOv4. From the results, we can see that ECA has better performance than some more popular methods. Although recall here is 0.08% lower than the CA method, the other three indicators are better than CA, and the comprehensive performance is better, which also verifies the effectiveness of the method in this paper.

Table 9. Comparison results of different attention mechanisms.

Attention mechanism	Precision	F1	Recall	MAP
SE	97.75%	97%	96.89%	98.74%
CBAM	97.79%	98%	97.43%	99.13%
CA	97.78%	97%	97.52%	99.21%
ECA	97.82%	98%	97.44%	99.26%

3.3.4. Comparison of detection images

In this section, we test YOLOv4 and improved YOLOv4 with the actual detection images to further judge the detection accuracy, and the test results are shown in Figure 16. We select images at different heights and scenes. Compared with YOLOv4, the ratio of improved YOLOv4 prediction frame to the real frame is higher, which is more appropriate for the actual range of the detection frame of RIFA nests. Moreover, improved YOLOv4 has few cases of missed detection and false detection, the robustness of recognition in complex scenes is strong, and relatively excellent detection results have been achieved.

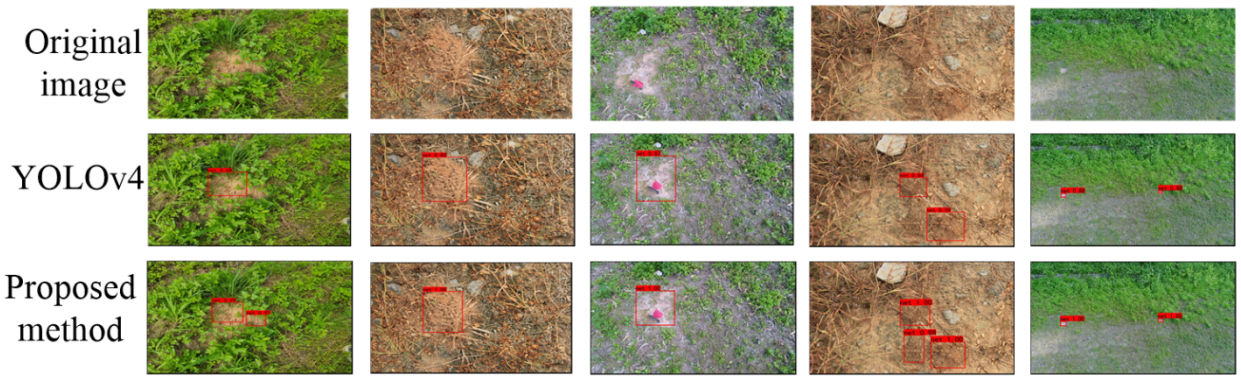

Figure 16. Comparison of detection images before and after YOLOv4 improvement.

3.4. Combination of algorithms and hardware

The Raspberry Pi 4B operating system used in this paper is Raspbian 10 Buster, which is developed based on armv71 Linux 5.10.103-v71+. The CPU uses the ARMv7 version, the processing frequency is 1.5 GHz, GPU is vc4drmfb, and the RAM space is 7847 MiB. After configuring the environment, we port our model to Raspberry Pi 4B, debug the code, and then assemble it with the UAV for detection in the airspace. To clearly display the results, we intercept a certain frame of image input from the video stream and detect it in the Raspberry Pi 4B. The terminal outputs the IOU value and the four-point coordinates of the detection frame. The detection process and results are shown in Figure 17. On the basis of ensuring that Raspberry Pi and the host are in the same local area network, we use Putty software for remote SSH wireless connection, WinSCP to transfer files and VNC Viewer to visualize the system interface. The image display interface is developed based on FFmpeg, which can make regular changes to the image. Since we use a lightweight model, it runs smoothly when detecting images on Raspberry Pi 4B, and the detected IOU and the host have the same effect, which can realize the efficient and flexible operation of the UAV.

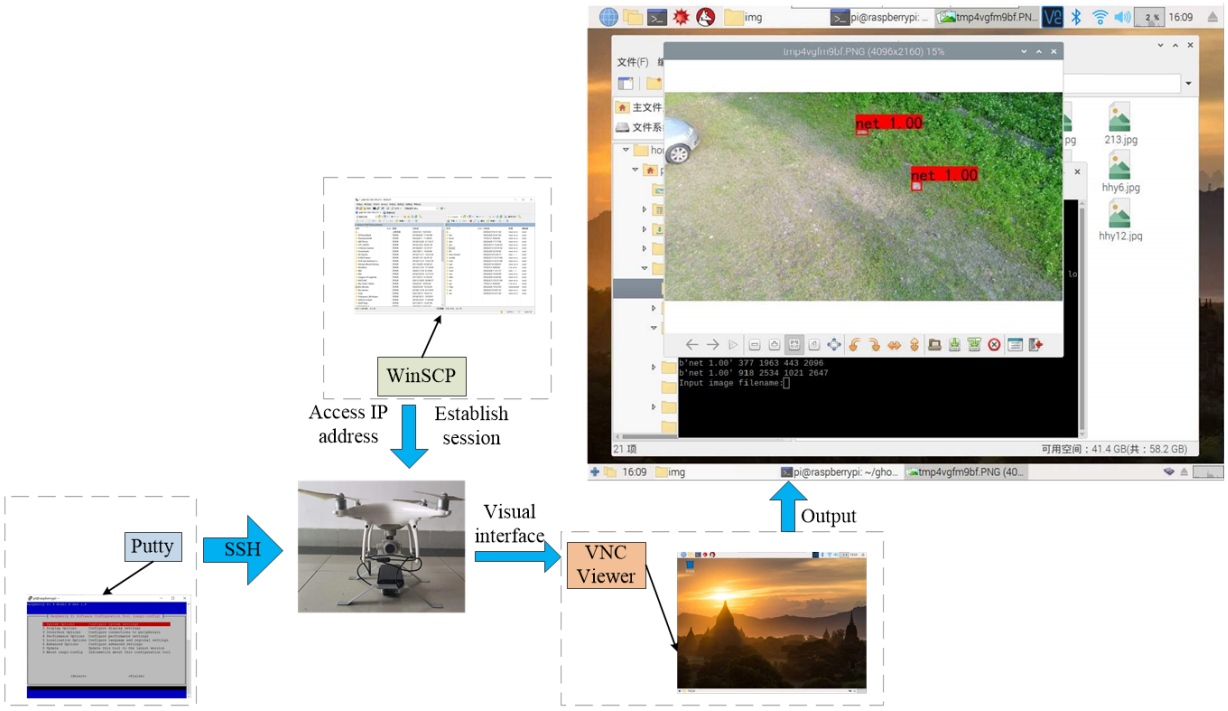

Figure 17. Detection process of modified UAV.

4. Conclusions

Aiming at the problems of poor robustness and low accuracy of RIFA nest detection in complex wild environments, this paper proposes a RIFA nest detection network based on improved YOLOv4. We add an improved GBN module to the network. Among them, the linear back-propagation characteristics of LeakyReLU in the negative direction can be used to speed up the convergence of

the model, and the cross-channel interaction strategy of the ECA attention mechanism can be combined to reduce the redundant computational cost. Furthermore, we optimize the Neck network using a copy-filled upsampling operation to strengthen the network's understanding of multi-layer semantic features. For the head network, we use Equalized Focal Loss to optimize to dynamically adjust the imbalance of positive and negative samples. Finally, improved YOLOv4 reduces the generation of redundant feature maps and improves the network's attention to key features. As a result, MAP of our network reaches 99.26%, which is 5.9% higher than the original network. The improved model achieves the highest accuracy and speed compared to some existing state-of-the-art models. Parameters are only 0.17 times of the previous one, which meets the conditions for porting in embedded devices. Meanwhile, this paper also makes the RIFAN dataset, which contains a variety of scenes at different heights, and uses the three-scale input method and data augmentation to improve the quality of the RIFAN dataset. Finally, we use a device composed of a UAV and Raspberry Pi 4B combined with a variety of software to conduct flying tests, and the results show excellent performance, which verifies the effectiveness of the proposed method.

Although the current work has a good detection effect on the spatial domain, the detection is still slow for high-resolution images, and there is room for further improvement in the number of effective features in the image. In future work, we will collect more high-resolution images of complex scenes, further improve the network backbone, reduce the amount of redundant parameters and increase the network's attention to effective features, so as to generate more application values for the control of RIFA.

Acknowledgments

This work was financially supported by the National Natural Science Foundation of China (Grant No. 61863011; No. 32071912), the Guangzhou Science and Technology Project (Grant No. 202102080337; No. 202002020016), the Project of Rural Revitalization Strategy in Guangdong Province (Grant No. 2020KJ261), the Natural Science Foundation of Guangdong Province (Grant No. 2020A1515010793), the Second Batch of Industry-Education Cooperation Collaborative Projects in 2019, Ministry of Education (Grant No. 201902062040), the Guangzhou Key Laboratory of Intelligent Agriculture (Grant No. 201902010081) and the Applied Science and Technology Special Fund Project, Meizhou, China (Grant No. 2019B0201005).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. L. Lv, Y. He, J. Liu, X. Liu, S. Vinson, Invasion, spread, biology and harm of Solenopsis invicta Buren (in Chinese), *Cant. Agric. Sci.*, **5** (2006), 3–11. https://doi.org/10.16768/j.issn.1004-874x.2006.05.001
2. W. B. Wu, L. Zhi, T. S. Hong, et al., Detection of Solenopsis invicta nest using spectrum analysis technology, *T. Chin. Soc. Agric. Eng. (Transactions of the CSAE)*, **29** (2013), 175–182.

3. R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, (2014), 580–587. https://doi.org/10.1109/CVPR.2014.81
4. R. Girshick, Fast R-CNN, in *2015 IEEE International Conference on Computer Vision (ICCV)*, (2015), 1440–1448. https://doi.org/10.1109/ICCV.2015.169
5. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39** (2016), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
6. K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, in *IEEE Transactions on Pattern Anal ysis and Machine Intelligence*, **42** (2017), 386–397. https://doi.org/10.48550/arXiv.1703.06870
7. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 779–788. https://doi.org/10.1109/CVPR.2016.91
8. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, SSD: single shot multibox detector, in *European Conference on Computer Vision*, (2016), 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
9. A. Roy, J. Bhaduri, Real-time growth stage detection model for high degree of occultation using DenseNet-fused YOLOv4, *Comput. Electron. Agric.*, **193** (2022), 106694. https://doi.org/10.1016/j.compag.2022.106694
10. A. Roy, R. Bose, J. Bhaduri, A fast accurate fine-grain object detection model based on YOLOv4 deep neural network, *Neural Comput. Appl.*, **34** (2022), 3895–3921. https://doi.org/10.1007/s00521-021-06651-x
11. M. O. Lawal, Tomato detection based on modified YOLOv3 framework, *Sci. Rep.*, **11** (2021), 1447. https://doi.org/10.1038/s41598-021-81216-5
12. D. Wu, S. Lv, M. Jiang, H. Song, Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments, *Comput. Electron. Agric*, **178** (2020), 105742. https://doi.org/10.1016/j.compag.2020.105742
13. O. Agbo-Ajala, S. Viriri., A lightweight convolutional neural network for real and apparent age estimation in unconstrained face images, *IEEE Access*, **8** (2020), 162800–162808. https://doi.org/10.1109/ACCESS.2020.3022039
14. A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al., MobileNets: efficient convolutional neural networks for mobile vision applications, *Comput. Sci.*, (2017), 1–9. https://doi.org/10.48550/arXiv.1704.04861
15. X. Zang，X. Zhou，M. Lin，J. Sun, ShuffleNet: an extremely efficient convolutional neural network for mobile devices, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2018), 6848–6856. https://doi.org/10.1109/CVPR.2018.00716
16. M. Tan, Q. Le, EfficientNet: rethinking model scaling for convolutional neural networks, in *Proceedings of the 36th International Conference on Machine Learning*, (2019), 6105–6114. https://doi.org/10.48550/arXiv.1905.11946
17. F. Zhang, Z. Chen, R. Bao, C. Zhang, Z. Wang, Recognition of dense cherry tomatoes based on improved YOLOv4-LITE lightweight neural network (in Chinese), *Trans. Chin. Soc. Agric. Eng.*, **37** (2021), 270–278. https://doi.org/10.11975/j.issn.1002-6819.2021.16.033

18. M. Togacar, B. Ergen, Classification of cloud images by using super resolution, semantic segmentation approaches and binary sailfish optimization method with deep learning model, *Comput. Electron. Agric.*, **193** (2022), 106724. https://doi.org/10.1016/j.compag.2022.106724
19. X. Wang, X. Zhuang, W. Zhang, Y. Chen, Y. Li, Lightweight real-time object detection model for UAV platform, in *2021 International Conference on Computer Communication and Artificial Intel ligence (CCAI)*, (2021), 20–24. https://doi.org/10.1109/CCAI50917.2021.9447518
20. L. Yu, E. Yang, C. Luo, P. Ren, AMCD: an accurate deep learning-based metallic corrosion detector for MAV-based real-time visual inspection, *J. Ambient Intell. Hum. Comput.*, **2021** (2021), 1–12. https://doi.org/10.1007/s12652-021-03580-4
21. Y. Cheng, T. Zheng, Binocular visual obstacle avoidance of UAV based on deep learning (in Chinese), *Elect. Opt. Control*, **10** (2021), 31–35. https://doi.org/10.3969/j.issn.1671-637X.2021.10.007
22. V. Gonzalez-Huitron, j. León-Borges, A. E. Rodriguez-Mata, L. Amabilis-Sosa, B. Ramírez-Pereda, H. Rodriguez, Disease detection in tomato leaves via CNN with lightweight architectures implemented in raspberry Pi 4, *Comput. Elect. Agri.*, **181** (2021), 105951. https://doi.org/10.1016/j.compag.2020.105951
23. S. Yun，D. Han，S. J. Oh，S. Chun, J. Choe, Y. Yoo, Cutmix: regularization strategy to train strong classifiers with localizable features, in *Proceedings of the IEEE International Conference on Computer Vision*, (2019), 6023–6032. https://doi.org/10.1109/ICCV.2019.00612
24. Y. Ma, X. Cai, F. Sun, Towards no-reference image quality assessment based on multi-scale convolutional neural network, *Comput. Model. Eng. Sci.*, **123** (2020), 201–216. https://doi.org/10.32604/cmes.2020.07867
25. A. Bochkovskiy, C. Y. Wang, H. Y. M. Liao, YOLOv4: Optimal speed and accuracy of object detection, *preprint arXiv: 10934*, 2004.
26. Y. Wu, K. He, Group normalization, in *European Conference on Computer Vision*, **128** (2020), 742–755. https://doi.org/10.1007/978-3-030-01261-8_1
27. K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, C. Xu, GhostNet: more features from cheap operations, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), 1577–1586. https://doi.org/10.1109/CVPR42600.2020.00165
28. M. Wang, S. Jiang, J. Wu, C. Wang, Research on image defogging algorithm based on improved generative antagonistic network, *J. Changchun Univ. Sci. Technol.*, **44** (2021), 93–99.
29. Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, ECA-Net: efficient channel attention for deep convolutional neural networks, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. https://doi.org/10.1109/CVPR42600.2020.01155
30. B. Li, Y. Q. Yao, J. Tan, G. Zhang, F. Yu, J. Lu, Equalized focal loss for dense long-tailed object detection, preprint, arXiv: *2201.02593*.