remote sensing

MDPI

Article

Rapid Target Detection of Fruit Trees Using UAV Imaging and Improved Light YOLOv4 Algorithm

Yuchao Zhu [1], Jun Zhou [2], Yinhui Yang [1], Lijuan Liu [1], Fei Liu [2] and Wenwen Kong [1,*]

[1] College of Mathematics and Computer Science, Zhejiang A & F University, Hangzhou 311300, China
[2] College of Biosystems Engineering and Food Science, Zhejiang University, Hangzhou 310058, China
* Correspondence: wwkong16@zafu.edu.cn

Abstract: The detection and counting of fruit tree canopies are important for orchard management, yield estimation, and phenotypic analysis. Previous research has shown that most fruit tree canopy detection methods are based on the use of traditional computer vision algorithms or machine learning methods to extract shallow features such as color and contour, with good results. However, due to the lack of robustness of these features, most methods are hardly adequate for the recognition and counting of fruit tree canopies in natural scenes. Other studies have shown that deep learning methods can be used to perform canopy detection. However, the adhesion and occlusion of fruit tree canopies, as well as background noise, limit the accuracy of detection. Therefore, to improve the accuracy of fruit tree canopy recognition and counting in real-world scenarios, an improved YOLOv4 (you only look once v4) is proposed, using a dataset produced from fruit tree canopy UAV imagery, combined with the Mobilenetv3 network, which can lighten the model and increase the detection speed, combined with the CBAM (convolutional block attention module), which can increase the feature extraction capability of the network, and combined with ASFF (adaptively spatial feature fusion), which enhances the multi-scale feature fusion capability of the network. In addition, the K-means algorithm and linear scale scaling are used to optimize the generation of pre-selected boxes, and the learning strategy of cosine annealing is combined to train the model, thus accelerating the training speed of the model and improving the detection accuracy. The results show that the improved YOLOv4 model can effectively overcome the noise in an orchard environment and achieve fast and accurate recognition and counting of fruit tree crowns while lightweight the model. The mAP reached 98.21%, FPS reached 96.25 and F1-score reached 93.60% for canopy detection, with a significant reduction in model size; the average overall accuracy (AOA) reached 96.73% for counting. In conclusion, the YOLOv4-Mobilenetv3-CBAM-ASFF-P model meets the practical requirements of orchard fruit tree canopy detection and counting in this study, providing optional technical support for the digitalization, refinement, and smart development of smart orchards.

Keywords: tree detection; YOLOv4; attention mechanism; lightweight; feature fusion

Citation: Zhu, Y.; Zhou, J.; Yang, Y.; Liu, L.; Liu, F.; Kong, W. Rapid Target Detection of Fruit Trees Using UAV Imaging and Improved Light YOLOv4 Algorithm. *Remote Sens.* **2022**, *14*, 4324. https://doi.org/10.3390/rs14174324

Academic Editors: Gemine Vivone and Liang-Jian Deng

Received: 17 August 2022
Accepted: 28 August 2022
Published: 1 September 2022

1. Introduction

Fruit is an essential food in people's lives, and the fine planting of orchards is essential to fruit yield and quality. The detection and counting of fruit trees is a necessary part of the excellent planting of orchards, which is related to the planning of planting density and the detection of fruit trees growing in the orchards, and is an essential part of the intelligent orchard. At present, in the orchards in the study area, the detection and counting of fruit trees mainly relies on manual observation. However, as orchards are widely distributed, vast, and possess complex and diverse terrain, manual labor is inefficient and costly in terms of work and time. It can be seen that it is difficult to meet the needs of real-time operation, speed, and simplicity using the current methods of fruit tree detection and counting. Therefore, the study of lightweight, real-time and rapid detection and counting

methods for fruit trees in natural environments is of great importance for the refinement of orchard planting and the construction of intelligent orchards.

In recent years, the rapid development of small UAVs has provided new ideas [1] for the performance of crop surveys and new opportunities [2] for the use of imagery to assess agricultural park experiments due to advantages such as simple operation, the saving of human and material resources, rapid acquisition and high resolution, along with advantages [3] such as small platform, low cost, high imaging resolution, ease of operation, flexibility and wide range of applications compared to traditional satellite images. At low flight altitudes, the images are not affected by cloud cover, and the resolution of images obtained with HD cameras can reach the centimeter level [4], thus leading them to be increasingly used in fields such as agricultural monitoring and precision agriculture, making up for the lack of manual ways to obtain images of orchards [5].

Fruit tree canopy recognition and counting methods can be divided into two main types: one based on traditional computer vision algorithms and the other based on the currently hot deep learning method. Currently, with the rapid development of machine vision technology, LIDAR [6], thermal mapping [7], and high-resolution RGB images have achieved good detection results in crop detection. Among them, image processing and feature extraction are gradually becoming key techniques for tree canopy recognition; some scholars have successfully detected targets using target segmentation counting in traditional algorithms, such as morphology [8], notch matching [9], and watershed algorithms [10] or off-the-shelf software for direct detection, etc. However, the recognition and counting of fruit tree canopies in natural environments still presents significant challenges; in addition, problems such as adhesion and occlusion between fruit trees and the color of the canopy compared to weeds in the background severely limit the accuracy of fruit tree canopy recognition and counting. Cao et al. [11] used UAV images to produce DEM, DSM, and DOM to extract row trees based on the neighborhood maximum filtering method, and the plant count extraction rate reached over 95%. He et al. [12] used the maximum local method and multi-scale segmentation algorithm to extract the number of plants from coniferous and broad-leaved forests. The overall accuracy was around 90%. Teng et al. [13] used the iterative H-minima improved watershed algorithm for broadleaf forest canopy extraction, and the F-measure of this algorithm for extracting the canopy of broadleaf forest with a more regular shape was 92.71%. However, canopy detection by the algorithm mentioned in the above study is affected by the contour shape, texture features, and color features of the target. For example, the morphology-based segmentation method requires a high degree of morphological specificity of the detection target, and the tree crowns in this study have very different morphologies, along with problems such as mutual adhesion and occlusion; in addition, the concave point matching algorithm requires a smoother edge of the detection object, but the edges of most tree crowns are not smooth, and even if a series of binarized images of the tree crowns are targeted for when using the watershed algorithm, it is difficult to calculate the local extremes of the image due to the presence of a lot of texture in the canopy itself, resulting in frequent under- or over-segmentation; when using color features in different color spaces for recognition [14], it is not easy to distinguish between the canopy and the weeds in the background, and the canopy color also changes with the growth of the fruit trees. In summary, it is difficult to apply traditional computer vision algorithms in these scenarios, and therefore the detection and counting of fruit tree canopies in natural environments still faces great challenges.

In recent years, deep learning has been widely used in the field of pattern recognition, and it has achieved a high level of success in many areas such as computer vision, image analysis, and multimedia applications [15]. Unlike traditional algorithms, deep learning learns features automatically rather than manually finding the suitable algorithm based on the parts. Target detection algorithms based on convolutional neural networks can be classified into two categories: the first is two-step target detection, where candidate regions are first generated through the network and then put into a convolutional machine model for classification. For example, in 2014, Ross B. Girshick proposed Regional

Convolutional Neural Networks (RCNN), and later Fast R-CNN [16], Faster R-CNN [17], R-FCN [18], Mask R-CNN [19], etc. The second is single-step target detection; in 2016, Redmon J et al. proposed the YOLO network; in addition to the low accuracy of target detection in YOLO [20,21] network, researchers have successively proposed SSD, YOLOv2, YOLOv3 [22], YOLOv4 [23,24], YOLOv5, etc. Huang et al. [25] used orthophotos obtained from UAVs to identify tree crowns by improving the Faster R-CNN target detection method, using ResNet101 to replace the VGG16 base network, and by improving the feature pyramid, the accuracy of tree crown extraction reached 92.92%. Jing et al. [26] used a faster regional convolutional neural network Faster R-CNN model for ground apple tree detection and counting, and compared with the traditional Hough transform and watershed algorithm, the Faster R-CNN model achieved an average accuracy of 95.53%. Chen et al. [27] used the improved YOLOv3 model for spruce counting based on UAV images, and achieved fast and accurate counting of spruce numbers by adding a dense connection module and an over the module to the trunk extraction network Darknet-53. Zheng et al. [28] proposed an improved YOLOv4-tiny-based single-wood detection method, which finally achieved a performance optimization of nearly 46.1% compared to traditional methods such as the local maximum method and the watershed algorithm; it also outperformed novel methods such as the Chan-Vese model and the template matching method by nearly 26.4% compared to them. The above study shows that deep learning is highly robust for tree canopy detection.

YOLO algorithm is a high-precision target detection method, and with the development of the YOLO algorithm, YOLOv4 has received increasing attention. However, the core of the detection based on the CNN method is based on the region proposal method, that is, first select the sliding window or extract the proposal to train the network, and then classify it in the proposed region [29]. Furthermore, YOLOv4 is currently mostly used in laboratory environments, and the limitation of this method is that the background region is often mis-detected or missed. Wu et al. [30] successfully detected apple blossoms using YOLOv4, but we have found that YOLOv4 is not able to achieve proper bounding box localization, and it is difficult to distinguish overlapping detection objects. The emergence of attention mechanisms makes it possible to effectively address these problems by processing information by focusing only on regions of information that are beneficial to the task implementation and filtering out secondary information to improve the model, which has been employed in image classification [31], image segmentation [32] and image detection [33]. In deep learning, the commonly used attention mechanisms are the channel attention mechanism, the spatial attention mechanism, and the dual attention mechanism for both space and channel; of these, SENet (Sequeeze and Excitation Net) [34] is a typical channel attention mechanism, STNet (Spatial Transformer Network) [35] is a typical spatial attention mechanism, and CBAM (Convolutional Block Attention Module) [36] is a typical dual attention mechanism. At the same time, YOLOv4 itself has a large number of parameters, leading its detection model to be large in size and slow in speed, making it difficult to use on a large scale. The emergence of the Mobilenet [37] series of lightweight networks can effectively solve this problem, as the Mobilenet series of algorithms turn traditional convolution into deeply separable The Mobilenet algorithm transforms the traditional convolution into a depth-separable convolution, which greatly reduces the computation time and the number of parameters without compromising accuracy. Since YOLOv4 uses the PANet [38] structure, which is insufficient for multi-scale feature fusion, we need to enhance the feature fusion capability of the network to improve the perceptual capability of the model and improve the detection accuracy of the model. Therefore, there are still issues worth exploring for YOLOv4-based fruit tree canopy detection.

This study takes fruit trees in a natural environment acquired through UAV remote sensing as the research object, proposes an improved YOLOv4 method for fruit tree canopy detection, uses the lightweight network Mobilenetv3, and introduces the attention mechanism and adaptive feature fusion ASFF, thus optimizing the generation of preselected boxes, making the model not only lightweight, but also improving the canopy detection

accuracy for digitalization, satisfying the detection and counting of fruit trees in the orchard for the refinement and intellectual development of smart orchards. This study looks at the feasibility of using deep learning methods for fruit tree detection and counting in natural environmental conditions and verifies that the proposed model can quickly and accurately identify fruit trees against complex backgrounds. The objectives of this study are to: (1) Lighten the model by using a lightweight network to replace the backbone network of YOLOv4; (2) Improve the robustness of the model by using a two-channel attention mechanism to eliminate background noise; (3) Enhance the feature fusion and feature extraction capabilities of the model using an adaptive feature fusion module to improve the recognition accuracy of the model; (4) Achieve accurate fruit tree counting by marking the recognized images with colored boxes.

2. Materials and Methods

2.1. Data Collection

2.1.1. Overview of the Study Region

The study area (121.52′5″E, 29.18′51″N) is located in the "Red Beauty" orchard in Xiangshan County, Ningbo City, Zhejiang Province, China. This area has a subtropical maritime monsoon climate with abundant light, an average annual temperature of 16–17 °C, and abundant rainfall (average annual precipitation over 1400 mm). However, typhoons are frequent and often accompanied by violent storms, and the orchard needs intelligent management. The orchard is uneven and has a complex tree species composition; it is also a large orchard, and the growth of the trees varies greatly and is irregularly distributed. The fruit trees are located against a complex background, with canopies that are connected and shaded, and are not easily distinguishable from each other due to their similar color. It would be costly and time-consuming to rely on traditional fruit farmers for manual measurement and management.

2.1.2. Data Acquisition

The image data were collected when the fruit trees were ripe (10 November 2021). The drone platform is the DJI Royal MAVIC2 Pro (Da Jiang Innovations, Inc., Shenzhen, Guangdong, China), which weighs 907 g, including an overall flight battery weight of 240 g, measures 198 mm (L) × 83 mm (W) × 83 mm (H), and has an endurance of 20 min. It is equipped with a one-inch CMOS RGB camera sensor to acquire image data, a single-lens visible sensor with 20 million effective pixels and a single image resolution of 4000 × 3000, and is combined with ground software DJI Pilot for route planning. To prevent distortion caused by weather conditions, images were acquired between 11:00 and 14:00 during which time the sun was shining and stable, and the wind was light. The camera was positioned perpendicular to the ground to capture a frontal view of the experimental plot, ensuring a 75% overlap between the frontal and lateral views to achieve good image stitching performance. The images were stitched together using DJI Terra software (Da Jiang Innovations, Inc., Shenzhen, Guangdong, China) to obtain an ortho-image of the test site. After performing cropping and other methods, the experimental area was obtained as shown in Figure 1.

2.1.3. Dataset Production

In this study, sample images of fruit trees with different sizes, growth conditions, and shading levels were screened, and 600 images with a resolution of 512 × 512, 600 images with 768 × 768, 90 images with 1024 × 1024, and 90 images with 2048 × 2048 were obtained following screening. To improve the generalization ability and robustness of the network model, this study used data enhancement to increase the number of samples to prevent the network from overfitting due to the lack of training samples. Data enhancement was carried out using image flip, image brightness adjustment, and noise addition to obtain an enhanced data set of 3000 samples, which was then divided into a training set of 2430, a validation set of 270, and a test set of 300. The targets were then annotated using

LabelImg (App version: 1.8.5) (https://github.com/heartexlabs/labelImg, accessed on 16 August 2022) to form an XML file in PASCAL VOC format.

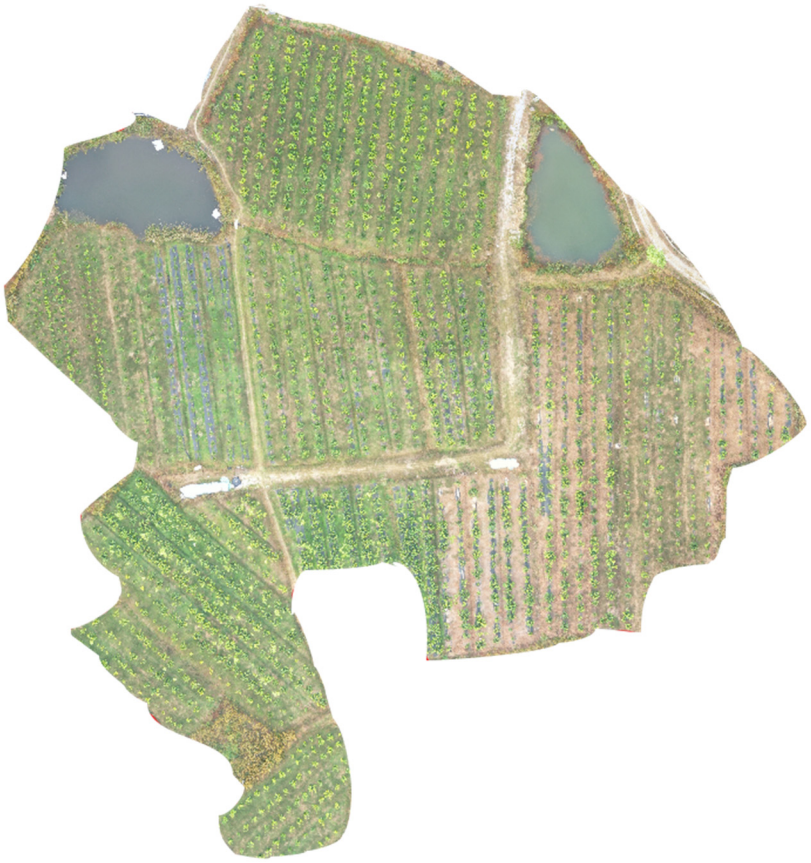

Figure 1. Cropped orthophoto of the study area.

2.2. Methods

2.2.1. Test Environment and Parameter Settings

The fruit tree detection models were all trained in the TensorFlow framework. The hardware environment was an AMD (Advanced Micro Devices, Santa Clara, CA, USA) Ryzen 5 3600X 6-Core Processor with 32 GB of running memory and an NVIDIA GeForce RTX2070S graphics card. The software environment was Windows 10 with network installations of Python 3.7, Cuda 10.0, and Cudnn 7.4. The image input size was 416 × 416 px. A freeze training strategy was used for training. First, the backbone network parameters were frozen for 75 training steps, with eight images per batch and the learning rate set to 0.001, and then unfrozen for 75 training steps, with four images per batch and the learning rate set to 0.0001, for a total of 150 iteration steps. The IoU threshold sets to 0.5. Choosing a suitable learning rate accelerates the convergence of the model. This prevents it from oscillating around the minimum value. This study uses the cosine annealing strategy and the hot restart method provided by the TensorFlow framework to enable the training to go beyond the local optimum and approach the global optimum.

2.2.2. Overview of the YOLOv4 Model

YOLOv4 uses new techniques based on YOLOv3 to improve object detection accuracy and speed, allowing it to achieve high-accuracy detection in real time. However, like fruit trees, the object of this study is not only a tiny target, but also exists against a variety of complex backgrounds; the canopy and the background of weeds, cooking smoke, flowers, etc., obscure each other. As the network layers deepen in the forward propagation process, the features of small targets partially disturbed by obscuring and similarly shaped and colored weed backgrounds are further weakened; thus, the detailed features of these small

targets gradually disappear in the subsequent network propagation process, causing false detections or missed detections. Therefore, to better put the model into practice in the future, it is essential to address the interference susceptibility of small target detection against complex backgrounds.

To solve the above problems, this study adopts the lightweight network Mobilenetv3 to replace YOLOv4's original backbone feature extraction network, the CSPDarknet53 network; meanwhile, the traditional convolution is replaced by a deep separable volume machine. In addition, the attention mechanism is introduced into the neck network PANet, which applies the attention mechanism to the feature layer output from the backbone network, and also applies the attention mechanism to the upsampling module of the neck network. After the PANet module, the ASFF module is added to enhance the ability of neck network feature fusion. The improved YOLOv4 network structure is shown in Figure 2. It is mainly composed of the MobilenetV3 backbone feature extraction network, a feature fusion module, and a prediction module.

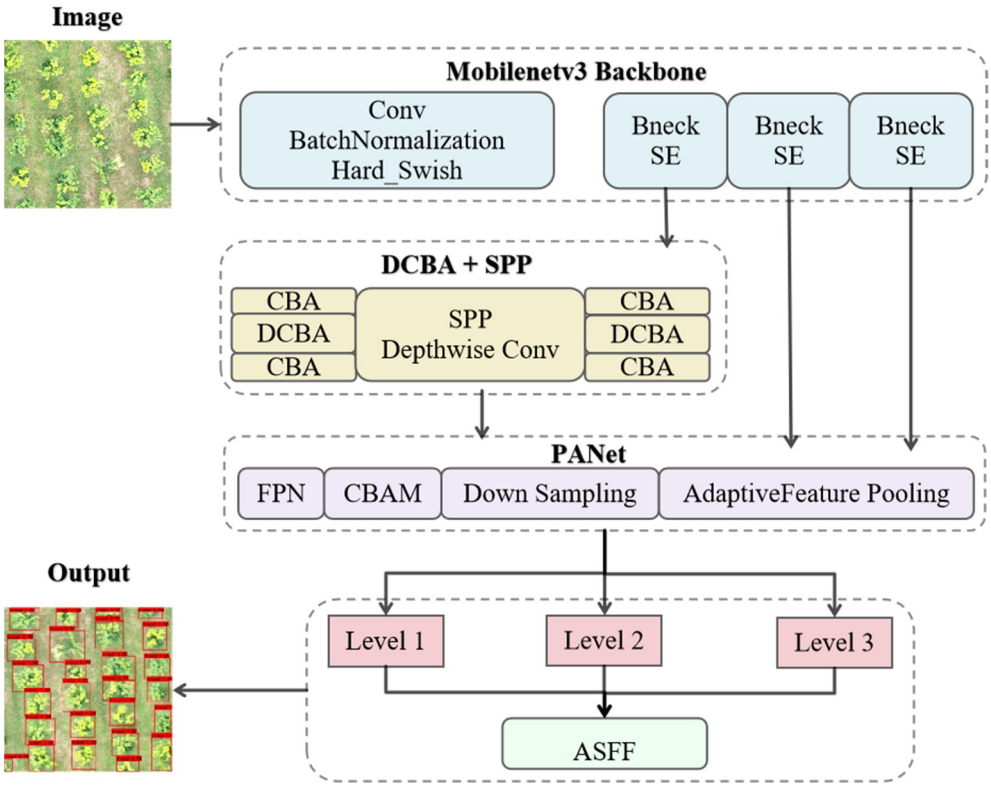

Figure 2. Overview of the YOLOv4 Model.

The specific structure of the CBA and DCBA modules describes in Figure 2. An image with a size of 416 × 416 × 3 is input into the detection model, which eventually yields a prediction network of 13 × 13, 26 × 26, and 52 × 52. The prediction module of the model makes a judgment and then obtains a prediction frame by adjusting the a priori frame.

2.2.3. Lightweight Backbone Network

When choosing a network structure for deep learning, there are many aspects to consider, such as accuracy, real-time operation, speed of operation, etc. It should also take into consideration its specific application scenarios or goals, such as, in the case of target detection scenarios based on UAV remote sensing, ensuring a lightweight model for being embedded in portable devices. In the traditional YOLOv4 backbone feature extraction network, point-to-point convolution is performed between the image and the filter, leading to high computational effort; therefore, we introduce depth-separable convolution to overcome this problem. The standard convolution operation is a convolution operation

using multiple convolution kernels on numerous input channels, with the parameter computation C shown in Equation (1), where D_K is the size of the convolution kernel, M is the number of input channels, N is the number of convolution kernels, and D_F is the size of the input image. Deeply separable convolution decomposes a complete convolutional block into two blocks, first using a convolutional kernel on M input channels and then adjusting the number of output channels by pointwise convolution, i.e., using N 1×1 convolution kernels, with the parameter computation C′, as shown in Equation (2). As shown in Equation (3), the number of parameters and operations in the network is significantly reduced, and the processing efficiency increases with minimal impact on the model accuracy. In this study, MobileNetv3, with its lightweight architecture, is used to replace the original CSPDarknet53 network. Howard et al. [39] build on the previous MobilenetV1 and MobileNetV2 and use NetAdapt's network architecture search method to optimize Mobilenet. In addition, MobilenetV3 uses the h-swish activation function to reduce the number of training parameters and the number of operations, as shown in Equation (4), thus reducing the number of memory accesses, reducing the complexity of the model, and improving performance.

$$C = D_K \times M \times N \times D_F \tag{1}$$

$$C' = D_K \times M \times D_F + M \times N \times D_F \tag{2}$$

$$\frac{C'}{C} = \frac{1}{N} + \frac{1}{D_K} < 1 \tag{3}$$

$$h - swish(x) = x \times \sigma(x) \tag{4}$$

$$\sigma(x) = \frac{ReLU6(x+3)}{6} \tag{5}$$

$\sigma(x)$ denotes a segmented linear simulation function, as shown in Equation (5).

Figure 3 shows that MobilenetV3 contains a core module called BNeck, an inverse residual structure block, a depth-separable convolution block, a SE attention module, and two activation functions, ReLU and h-swish. The inverse residual structure block connects input and output features on the same channel via an inverted residual join, first using 1×1 convolution for up-dimensioning and then the following operation with residual edges. In addition, it contains a SENet attention module that adjusts the weights of each channel so that training focuses more on the relevant features of each channel, boosting useful features and suppressing features of limited usefulness.

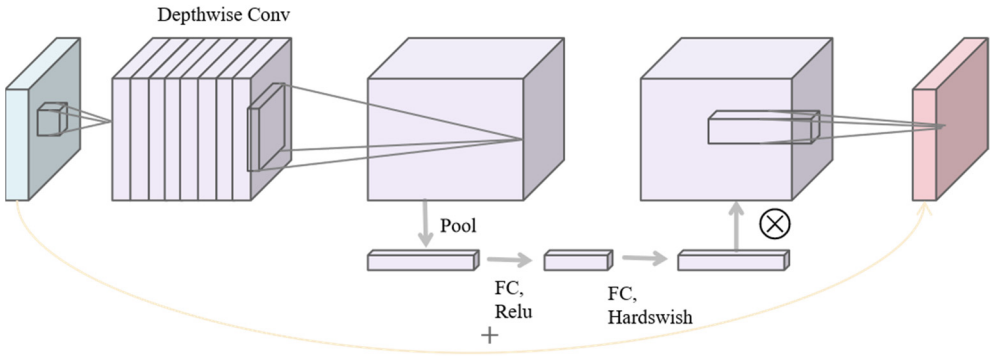

Figure 3. Schematic diagram of the BNeck structure.

As the use of deep separable convolution increases the number of layers in the network, if the network model is too deep, the gradient dispersion is likely to cause the model to fail to converge, so the residual structure introduces to improve this problem. The residual structure features an additional shortcut connection to achieve data superposition between inputs and outputs without increasing the number of parameters and complexity

of the model, yet avoiding the occurrence of gradient vanishing, thus improving the expressiveness of the model again while using a depth-separable volume machine. In summary, the use of Mobilenetv3 to replace the CSPDarkNet53 backbone network and the introduction of a depth-separable convolution block simplifies the network structure and reduces computational memory consumption, resulting in a significant improvement in the training efficiency and detection speed of the algorithm model. It meets the requirements of smart orchards for fruit tree detection and counting methods to be lightweight and fast.

2.2.4. Channel Attention Module and Spatial Attention Module

The YOLOv4 algorithm treats the target detection process as a regression problem, which does not distinguish well between the foreground and background areas of the input image, leading to missed or false detections. In order to identify the target fruit trees, because of the small number of pixels and the complex background, some of the information about the fruit trees is obscured by other vegetation or canopy. Therefore, this study enhances the feature representation of the target by introducing an attention mechanism. In order not to increase the depth of the network, this study only replaces the residual connections in the enhanced feature extraction network by filtering the passing features with different weights so that the network values the channels with high weights and the information retained during residual fusion is more conducive to training loss reduction. For the target detection task, if the same amount of attention is paid to each feature map at the beginning of training, this can increase the time required for the network to converge. Different attention mechanism modules can all improve the accuracy of target detection. The best-performing approach is the one in which the channel attention mechanism in CBAM is directly connected to the spatial attention mechanism. The CBAM module adds a global maximum pooling operation to the channel attention module, thus compensating for some of the information lost due to global average pooling.

CBAM is a dual-dimensional attention mechanism based on the channel and spatial attention mechanisms to extract features. Figure 4 shows the network structure, consisting of CAM (Channel Attention Module) and SAM (Spatial Attention Module). CAM first compresses the input feature map of $H \times W \times C$ into a one-dimensional vector of $1 \times 1 \times$ C, bypassing the input image F through a set of maximum pooling and average pooling layers; the pooled one-dimensional vector is passed into the Muti-Layer Perception (MLP), which consists of two fully connected layers and an activation function. After the first fully connected layer, the channel dimension downscales from C-dimension to C/r-dimension, and after the second fully connected layer, it is upscaled to C-dimension again. The summed elements are then passed through the Sigmoid operation to generate a one-dimensional vector Mc of $1 \times 1 \times$ C to obtain the output feature map F', as shown in Equation (6). Then, Mc is multiplied with the input feature map of $H \times W \times C$ as the input of the SAM module, and two sets of maximum pooling layers obtain two feature maps of $H \times W \times 1$ layers and average pooling layers, followed by splicing, dimensionality reduction operation, and Sigmoid operation to get the spatial attention feature map of $H \times W \times 1$. Finally, the attentional feature map with a size of $H \times W \times C$ is obtained by multiplying it with the input feature map of the SAM module F'', as shown in Equation (7).

The channel attention mechanism enhances the feature representation of obscured targets, while the spatial attention mechanism highlights regions of the feature map relevant to the current task.

$$F' = \mathrm{Mc}(F) \otimes F \tag{6}$$

$$F'' = \mathrm{Ms}(F') \otimes F' \tag{7}$$

The attention mechanism is a plug-and-play module, so it could theoretically be placed behind any feature layers, either in the backbone or the enhanced feature extraction network. However, as this study is based on migration learning, placing the attention mechanism module in the backbone network would result in the pre-training weights of the network not being available, so this study applies the attention mechanism module to the enhanced

feature extraction network. The CBAM module is inserted onto the two adequate feature layers extracted from the backbone network in this study. At the same time, it is added to the results after the upsampling module in the enhanced feature extraction network.

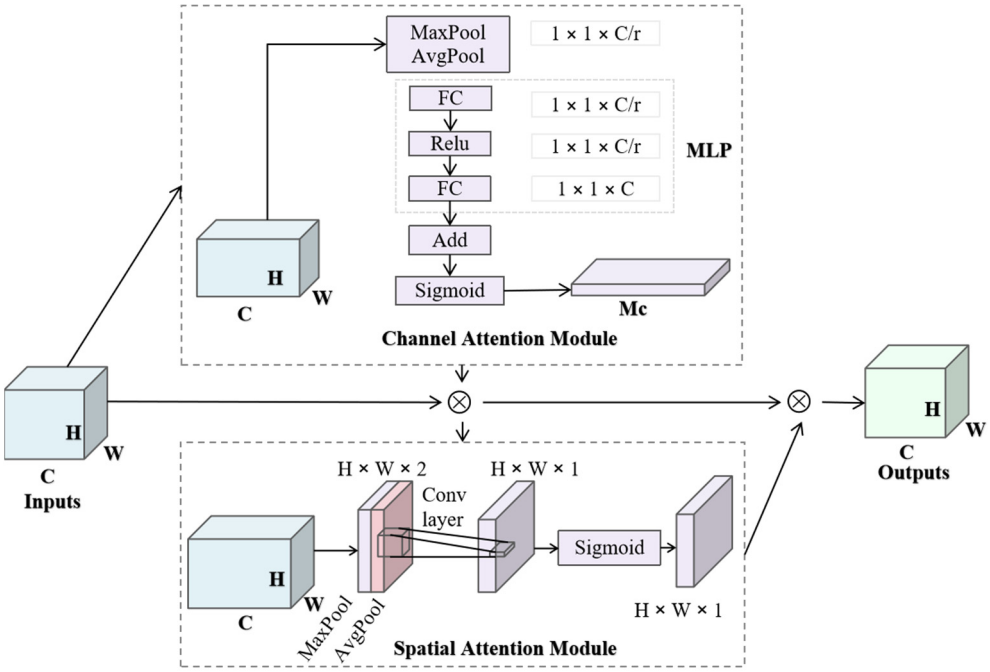

Figure 4. Convolutional block attention module.

2.2.5. Adding Feature Fusion Effects—Adaptive Spatial Feature Fusion

The original YOLOv4 enhanced feature extraction network is the PANet feature fusion network; when processing the FPN output feature map, the bottom-up result is used to fuse the higher-level information with the lower-level information, and YOLOv4 adds a bottom-up enhancement structure to the FPN, thus forming a two-way fusion of multi-scale feature maps. However, for the fruit tree canopy target in a complex context, the PANet feature fusion approach is still limited to the summation of feature maps at the same size. Still, it cannot fully use the features at different scales. Therefore, to solve this problem and make good use of the detailed information of the lower layer information and the semantic information of the higher layer information, this study introduces ASFF on PANet. The ASFF [40] structure learns each weighting coefficient for the three feature outputs of PANet by adaptive learning. Each layer's feature map is multiplied by the corresponding weight and then fused simultaneously.

Figure 5 shows the network structure diagram based on adaptive spatial feature fusion (ASFF). The input feature maps in the figure are the three feature maps output from the PANet network, which are Level1, Level2, and Level3. Then, the feature fusion operations of ASFF-1, ASFF-2, and ASFF-3 are performed on these three-layer feature maps, respectively, and finally, the fused feature maps are predicted. When using the ASFF module for feature fusion, each layer's size and number of channels are adjusted to be the same. The corresponding weight coefficients are calculated for adjusting feature maps. Then, each layer is multiplied by the corresponding weight coefficients of that layer and summed so that adaptive learning between features can be achieved. Taking ASFF-3 as an example, the feature fusion process can be represented as follows:

$$y_3 = \alpha_3 \times X_{1\to3} + \beta_3 \times X_{2\to3} + \gamma_3 \times X_{3\to3} \quad (8)$$

where y3 is feature map 3, $X_{1\to3}$, $X_{2\to3}$, $X_{3\to3}$ are the feature maps output when each layer feature map is adjusted to match the size and number of channels of the layer three feature

map, respectively; α_3, β_3, γ_3 are the weight coefficients learned when the three feature maps $X_{1\to3}$, $X_{2\to3}$, $X_{3\to3}$ are feature fused at the third layer, respectively. For α_3, β_3, γ_3, the weight values are firstly obtained by 1×1 convolution of the three feature maps $X_{1\to3}$, $X_{2\to3}$, $X_{3\to3}$, and then performing channel cascading, and finally using SoftMax to obtain the weight values in the range [0,1] and satisfy $\alpha_3 + \beta_3 + \gamma_3 = 1$.

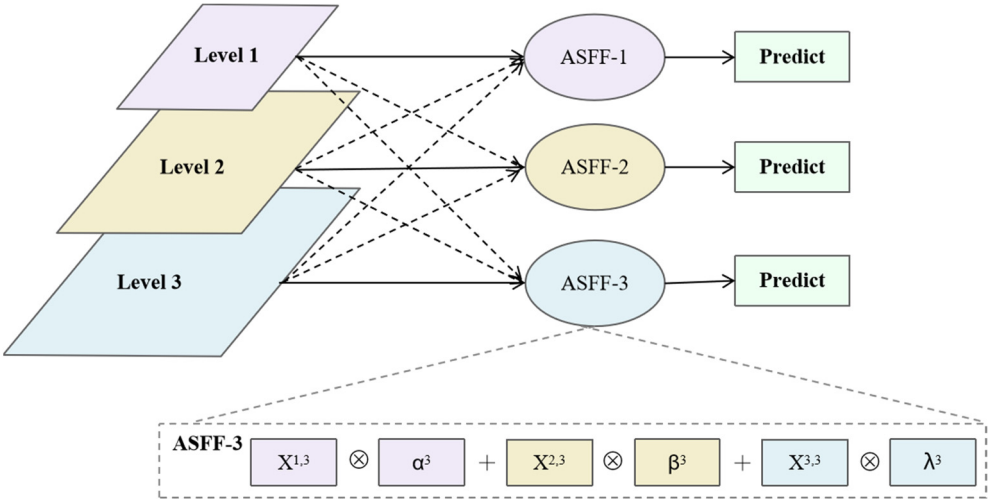

Figure 5. Adaptively spatial feature fusion.

As can be seen from the formula, in the ASFF module, each layer of the feature map is summed when it is fused, so the feature map size and the number of channels need to be adjusted in advance for each layer to be fused. Of course, as the required size and number of channels vary from layer to layer, the adjustment method is also different. As shown above, for ASFF-1, the feature map of each layer needs to be downsampled when it adjusts to the first level. For ASFF-3, the feature map of each layer needs to be upsampled when it adapts to the third level.

The ASFF module implements a fusion of the weight coefficients of each layer with the features multiplied and then added together, enabling the filtering of features from other layers to retain valid information for that layer, enabling the enhanced feature extraction network structure to extract features more hierarchically. In the fruit tree dataset in this study, there are differences in image sizes, where some of the targets occupy fewer pixels. Since the image size has to be resized to a uniform size when being re-inputted into YOLOv4, this causes the targets in the images to become smaller. Therefore, we need detailed features in the lower-level features to enhance the feature information of the small targets, and this is well met by introducing the ASFF structure.

2.3. Improved Generation Method of Pre-Selection Boxes

The K-means algorithm is based on iterative thinking to find the cluster centers. The steps of the algorithm are: select the initial K samples as the initial clustering centers of the algorithm, calculate the Euclidean distance to the K clustering centers for each sample in the dataset and assign it to the class corresponding to the clustering center with the smallest distance; then, for each new class, recalculate its clustering center and repeat the process until the clustering centers no longer change. It can be found that the selection of the anchor box is strongly related to the dataset, and the dataset in this study was made by ourselves, from the acquisition of the original data to the production of the dataset, and is a single type of dataset. However, if the anchor box generated by the K-means algorithm alone is relatively concentrated, the single K-means algorithm was shown to be ineffective for the generation of preselected boxes after experimental validation. It cannot achieve the advantage of the multi-scale output of the model. Therefore, this study combines the idea of linear scaling based on the K-means algorithm, and further optimizes the anchor box

generated by the K-means algorithm based on the mountain, thus stretching the anchor box and improving the detection accuracy of the model. The optimized anchor boxes are shown in Table 1.

Table 1. Optimized anchors of the different feature map.

Feature Map	13 × 13	26 × 26	52 × 52
Anchors	(16, 18) (94, 45) (70, 69)	(22, 44) (125, 115) (156, 162)	(279, 218) (355, 355) (684, 654)

3. Results

3.1. Evaluation Indicators

As the final model obtained in this study will be put into practical use in the future, not only does it need to have satisfactory accuracy, but the real-time nature of the model, the number of parameters, and the size of the model are also key evaluation metrics. All algorithms in this study were evaluated on the basis of mean average precision (mAP), F1-score, frames per second (FPS), and model size. Setting different IOU thresholds will result in different numbers of detected frames, where high thresholds result in a small number of detected frames and low thresholds result in a large number of detected frames. When the detected fruit tree canopy targets are small, detection may be missed if a larger threshold is set. Therefore, the threshold value set in this study is 0.5. The mAP is the average of the mean detection accuracy, which is the most important indicator of detection performance in target detection. The F1-score is the summed average of the model accuracy and recall, with a maximum of 1 and a minimum of 0. It provides a good assessment of the performance of the model, and the larger the F1-score, the better the model performance. The greater the frames per second (FPS), the more images can be detected per second, and the smoother the display. The model size is an important indicator of how lightweight the model is.

In addition, in this study, the number of tree crowns interpreted by visual interpretation was compared with the number of tree crowns extracted by each model and evaluated for accuracy. The Average Overall Accuracy (AOA) was used, which was calculated as follows:

$$AOA = \frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{|N_i - N|}{N} \times 100\%\right) \quad (9)$$

where N_i represents the total number of canopies obtained by identifying the ith sample map by the specified model; N represents the total number of canopies obtained by visually interpreting the ith sample map.

3.2. Results of Detecting in Different Models

This study is based on the YOLOv4 algorithm with improvements, which include the addition of the attention mechanism, the lightweight of the backbone feature extraction network, the addition of the ASFF module to the feature fusion module, and the improvement of the a priori preselected box, and the results are shown in Table 2. The model in Table 2 lists YOLOv4 and models based on YOLOv4 with different improvements, such as YOLOv4-Attention-Mobilenetv3, which is YOLOv4 after replacing the backbone network with Mobilenetv3 and applying the CBAM attention mechanism; YOLOv4-Attention-Mobilenetv3-ASFF-P replaces the backbone network with Mobilenetv3, applies the CBAM attention mechanism, adds the ASFF module and optimizes the YOLOv4 using the preselection box generation method. Then, the mAP, F1-score, FPS, and model size of each model are listed, respectively.

Table 2. The results of fruit tree detection based on different models.

Model	mAP	F1-Score	FPS	Model Size (MB)
YOLOv4	95.88%	88.08%	36.62	244.0
YOLOv4—SE	96.69%	88.54%	40.00	245.0
YOLOv4—CBAM	96.99%	88.72%	36.28	246.0
YOLOv4-Mobilenetv3	96.73%	88.42%	95.70	44.3
YOLOv4-Attention-Mobilenetv3	97.50%	90.08%	104.91	44.5
YOLOv4-Attention-Mobilenetv3-ASFF	97.68%	90.38%	101.58	50.7
YOLOv4-Attention-Mobilenetv3-ASFF-P	**98.21%**	**93.60%**	**96.25**	**50.7**

3.2.1. Result of Achieving Light Weight by Using Mobilenetv3

In this study, the original YOLOv4 backbone network CSPDarkent53 network was replaced with the lightweight network Mobilenetv3, which is denoted as YOLO-Mobilenetv3. As can be seen from the above table, when using the YOLO-Mobilenetv3 model for detection, the mAP improved by 0.85% compared to YOLOv4, and the F1-score improved by 0.34%. Most importantly, the FPS value and the model size of the model changed significantly: firstly, the FPS value jumped from 36.62 to 95.70, which is a significant increase of about 2.6 times; secondly, the model size decreased from 244 MB to 44.3 MB, which is about 0.18 of the original YOLOv4, which satisfies this study's goal of achieving light weight.

3.2.2. Result of Applying the Attention Mechanism

In this study, the attention mechanism module is inserted onto two adequate feature layers extracted from the backbone network. Also, in the enhanced feature extraction network, the attention mechanism module is added to the results after the upsampling module. The algorithm incorporating the SE attention mechanism (a common channel-based attention mechanism) in YOLOv4 is denoted as YOLOv4-SE, and the algorithm incorporating the CBAM attention mechanism is denoted as YOLOv4-CBAM. As can be seen in Table 2, compared to the traditional YOLOv4, the mAP of the network incorporating the SE attention mechanism and the CBAM attention mechanism increased by 0.81% and 1.11%, while the F1-score also increased by 0.46% and 0.64%, respectively. Furthermore, it can be seen that YOLOv4-CBAM performs better than YOLOV4-SE in terms of mAP. Other evaluation metrics, such as FPS and model size, only changed slightly.

3.2.3. Result of Applying the ASFF

After improving YOLOv4 using the CBAM attention mechanism module and the Mobilenetv3 lightweight network, this study added the ASFF module at the end of PANet to further enhance the feature fusion between the different layers. As can be seen from Table 2, the mAP value of the model after the addition of ASFF reached 97.68%, a significant improvement in detection accuracy, in addition to a small improvement in the F1-score of the model; the FPS value reached 101.58, and the model size, etc., was also reduced compared to the original YOLOv4, and compared to the YOLOv4-Attention-Mobilenetv3 model, the model size only increased slightly. It can be found that the model with the introduction of the ASFF module has improved in terms of mAP and F1-score performance.

3.2.4. Result of Optimizing Preselected Boxes

After optimizing the preselection box using K-means clustering combined with linear scaling, the model is denoted as YOLOv4-Attention-Mobilenetv3-ASFF-P. As can be seen from Table 2, after optimizing the preselection box, the mAP of the model reaches 98.21% and the F1-score value reaches 93.60%, both of which are the best among all models. It can be found that the model training after optimizing the pre-selected boxes can improve the detection accuracy and performance of the model.

To verify the detection capability of the YOLOv4 model modified by different strategies, fruit tree canopies were predicted in different scenes, and the results are shown in Figure 6. Three representative sample plots were selected: module a in Figure 6, from

left to right, shows scenes with high clarity in natural fields but with interference from people and other factors; the middle image shows scenes with severe adhesion between tree crowns, which are difficult to distinguish with the naked eye, and the right image shows scenes with a high number of tree crowns and average adhesion. The modules b, c, d, and e in Figure 6 show the results of the different modifications. As can be seen from the above figure, the detection of each scene becomes progressively better from b to e. In b, which is the original YOLOv4 detection model, it can be seen that problems such as missed detections, duplicate markings, and inappropriate box sizes exist and are the most serious. Finally, it can be seen that the YOLOv4-CBAM-Mobilenetv3-ASFF-P detection model has no problems with missed or duplicate marks, and the size of the marker box is appropriate. Combining the results in Table 2, it can clearly be concluded that the detection model with the optimized pre-selected box generation method has the best detection results.

3.3. Results of Counting in Different Models

To verify the counting ability of the YOLOv4 model modified by different strategies, 10 randomly selected sample images were counted using different models, and the number of tree crowns in the images was counted, with Yolov4 as model A, YOLOv4-CBAM-Mobilenetv3 as model B, YOLOv4-CBAM-Mobilenetv3-ASFF as model C, and YOLOv4-CBAM-Mobilenetv3-ASFF-P as model D. The counting results are shown in Table 3. P1-P10 in Table 3 are the 10 randomly selected sample images, and manual counting is the result after visual interpretation. The average overall accuracies of the four models are 77.06%, 85.01%, 92.61%, and 96.73%, respectively. The average overall accuracy of model D (YOLOv4-CBAM-Mobilenetv3-ASFF-P) is very impressive, although it has some deviations from the actual visual interpretation numbers. It can also be seen that the AOA progressively improves from model A to D. Therefore, the series of strategies we propose all have an improvement on the counting effectiveness of the model, leading to an increase in the robustness of the model.

Table 3. The results of fruit tree counts based on different models.

Counts	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	AOA
nums	13	10	21	11	23	20	31	73	71	55	/
Model A	9	8	26	10	14	19	33	53	50	34	77.06%
Model B	11	9	20	10	15	20	27	60	57	41	85.01%
Model C	12	9	21	10	20	20	29	69	63	49	92.61%
Model D	13	10	21	11	21	20	30	78	65	52	96.73%

(**a**)

Figure 6. *Cont.*

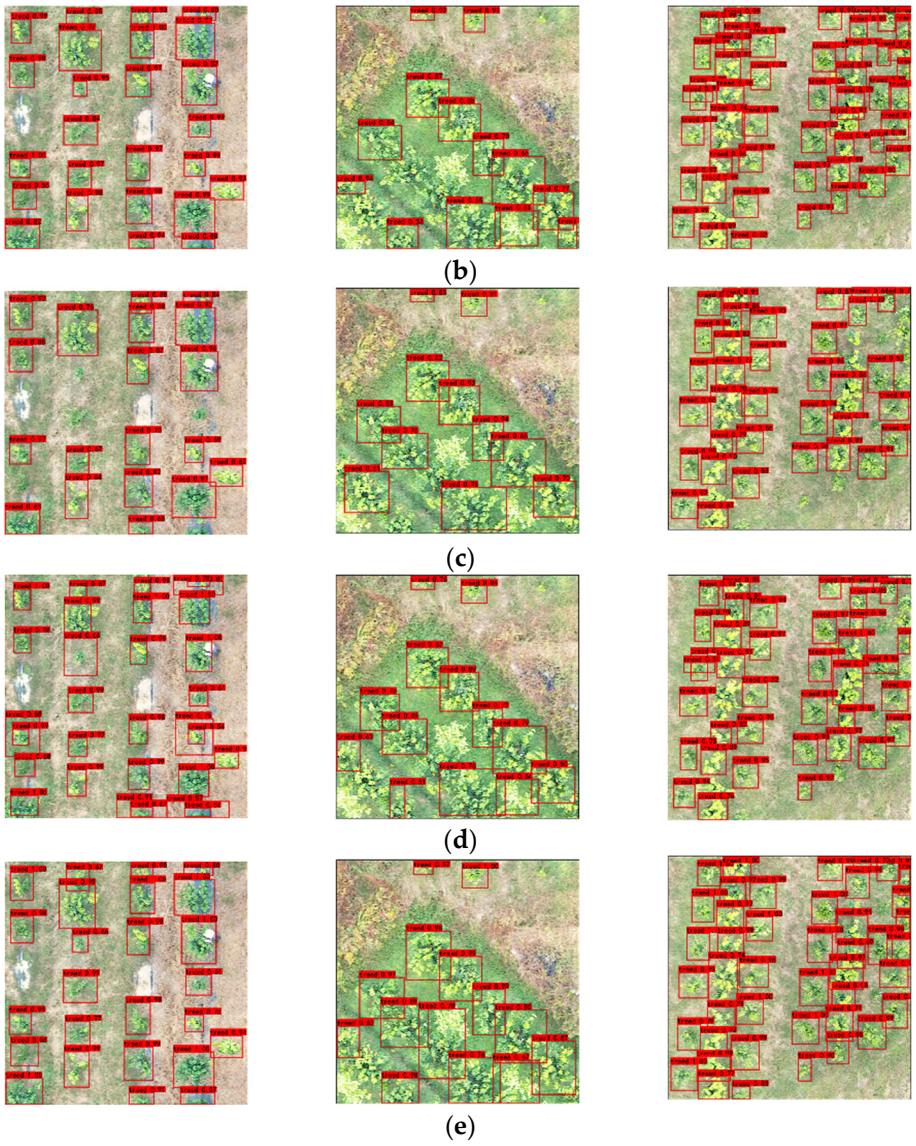

Figure 6. The recognition results of different models in different scenarios: (**a**) original images of different scenes, (**b**) detection results of YOLOv4, (**c**) detection results of YOLOv4-CBAM-Mobilenetv3, (**d**) detection results of YOLOv4-CBAM-Mobilenetv3-ASFF, (**e**) Detection results of YOLOv4-CBAM-Mobilenetv3-ASFF-P.

4. Discussion

This study builds a highly accurate, fast, and lightweight detection model based on an improved YOLOv4. In terms of dataset production, images containing different resolutions and including different complex backgrounds were constructed to improve the robustness of the model. With respect to model training, we adopted the training mode of cosine annealing, and employed the generation mode of the pre-selection box before training. Then, the main feature extraction network and the enhanced feature extraction network of the YOLOv4 network were improved. Finally, the trained model was tested in different scenarios, proving that the model is able to meet the requirements proposed in this study.

To embed recognition and counting models into mobile devices, an important goal of this study is to achieve lightweight models without losing much accuracy. As can be seen in the experimental results in Table 2, when replacing YOLOv4's backbone network CSPDarknet53 with the lightweight network Mobilenetv3, the FPS values of the model improve significantly, and the model size decreases significantly. We can also see that the mAP values of the model also improve slightly, compared to the decrease in accuracy that

occurs when Huang et al. [41] identified trees using the improved method of replacing YOLOv4's backbone network with Mobilenetv2 [42]; this is due to the presence of the SE attention mechanism module in the BNeck module in Mobilenetv3, which helps to improve the model's ability to provide detailed information. In addition, the model achieves an FPS of over 96, which is a significant reduction in detection time compared to Yang et al. [43], who employed CBAM-YOLOv4 to identify grape leaves (single image detection time was around 1 s), and well meets the demand for real-time detection in this study. In addition, the training parameters of the improved model were significantly reduced (from 6.44×10^7 to 1.18×10^7). Since more training parameters require more resources to be allocated to training, the computation time will be longer, and the configuration of the running environment will be more demanding. The size of the model obtained after training was also significantly reduced from 244 MB to about 50 MB. The improved lightweight model therefore not only improves computational efficiency but also optimizes the size of the model for mobile embedding and gives it the ability to run in low-configuration environments.

A comparison was made with the traditional YOLOv4 from the perspective of the attention mechanism module. It can be seen that both the SE attention mechanism and the CBAM attention mechanism have higher mAP values than the traditional YOLOv4, with small changes in FPS and model size. It can be concluded that the use of the attention mechanism can lead to an improvement in the detection accuracy of the network, as the attention mechanism allows the network to focus on what it needs to focus on more, improving the model's ability to perceive information. In addition, the model with YOLOv4-CBAM performs better, with a 1.11% improvement in mAP value; this is because CBAM has an additional spatial attention mechanism module compared to SE's channel-only attention mechanism, as not all regions in the image are equally important in contributing to the task, and only the task-related regions are worth caring about. The addition of this module helps the model to improve its ability to perceive location information and find the most important parts of the network for processing, thus improving the model's detection accuracy and counting ability.

From the experimental results, it can be observed that the mAP value is improved after the introduction of the ASFF module to the model, which is because of the three feature layers input to PANet, large targets in the image are detected at the top level and small targets at the bottom level; however, the layer-to-layer interaction exists only for upsampling and downsampling operations, and many of the top- and bottom-level features have not been utilized. Large targets in the image require larger perceptual fields and high-level semantic features, and small targets require fine-grained features in the underlying features to be discriminated; the introduction of the ASFF structure greatly enriches the model by adaptively learning the weight coefficients (the weight coefficients in ASFF are obtained by convolving the feature maps of each layer, and the corresponding weights are adaptively adjusted to smaller values for invalid feature maps, thus reducing the interference of invalid feature maps with target detection) for mapping and fusion of feature layers at each scale, fusing features from different layers together by learning the weight coefficients and filtering features from other layers to retain only the useful information in that layer. The ability to perceive the semantic information at the higher levels of the citrus canopy and the detailed information at the bottom layer is greatly enriched. Although the size of the model and the number of parameters became larger after the introduction of the ASFF structure, even though the model size and the number of parameters increased slightly, the overall optimization of the model achieved by using this structure was not affected, as this study focused more on the improvement of mAP, as well as the counting ability (mainly in AOA). From Tables 2 and 3, it can be seen that with the addition of the ASFF module, both mAP and AOA were significantly enhanced, further confirming the great improvement of ASFF for recognition and counting in this study.

As the label files in the dataset of this study were generated by manual annotation, the size is relatively concentrated, and most of the targets are small. Therefore, this study uses the K-means algorithm to optimize the generation of pre-selected boxes and uses

linear scale scaling for pre-selected box stretching on this basis. The experimental results show that the mAP values of the models trained using the optimized preselected boxes and the unoptimized preselected boxes are higher, at 98.21% and 97.68%, respectively. This is because the anchor frames optimized with the K-means algorithm and linear scale scaling can fit the training target better in multi-scale training and their generalization ability is better; larger anchor frames produce less loss and thus higher accuracy as they retain the feature information in the shallow network when transferring the feature information extracted from the shallow network to the deep network.

In this paper, the size of the model increased slightly after the addition of the CBAM attention mechanism, but the impact on real-time detection was not significant; to address this issue, a lighter CBAM model can be built to be embedded in the YOLOv4 network in the next step of research, which could improve the detection accuracy and lighten the model at the same time. Secondly, the lightweight network in this paper has only been implemented on the PC side, and has not yet been validated on the mobile side. In the future, we could study how to deploy the model to the mobile site or even embed it into the UAV for real-time detection. If the anchor box can be dynamically scaled in real time during the training process, the model's accuracy could be significantly improved, and accurate matching could be achieved. As the ASFF is inserted directly after the PANet structure, it does not make good use of the semantic information of the top-level features and the detailed information of the bottom-level features because of the top-down and bottom-up structures in the PANet structure itself. Therefore, in the future, changing the PANet structure could be considered to enable ASFF to make better use of information from the top and bottom layers.

5. Conclusions

In this study, we combined YOLOv4, a lightweight model, an attention mechanism, and feature fusion strategies to train, validate and test different models based on a dataset produced from fruit tree canopy UAV images, and compared the results and performance of the models with respect to recognition and counting under different strategies. Finally, a YOLOv4-Mobilenetv3-CBAM-ASFF-P approach was proposed, and a YOLOv4-Mobilenetv3-CBAM-ASFF-P model for fruit tree canopy recognition and counting was obtained. The key contribution is the good robustness of the model for both detection and counting. By introducing Mobilenetv3 as the backbone network, the YOLOv4 model is made to be more lightweight and computationally efficient; by incorporating the CBAM attention mechanism, the YOLOv4 model is able to focus more on important canopy features in the image and suppress unimportant features; by introducing the ASFF module, the fusion capability of the model's multi-scale features is enhanced, and the detection accuracy is further improved. Finally, the model is trained by optimizing the generation of pre-selected frames, combined with the learning strategy of cosine annealing, which speeds up the training speed of the model and improves the detection accuracy. The model achieved 98.21% mAP, 96.25 FPS, and 93.60% F1-score for canopy detection, a significant reduction in model size; and 96.73% AOA for canopy counting. This means that the model can effectively recognize and count fruit tree crowns, meeting the demand for lightweight, real-time and fast recognition and counting of fruit trees in the construction of smart orchards, and is portable and easy to apply, providing a solution and reference for the application of fruit tree crown recognition and counting models in practical scenarios.

Author Contributions: Conceptualization, W.K. and Y.Z.; methodology, W.K., Y.Z. and J.Z.; soft-ware, Y.Z. and J.Z.; validation, Y.Z., Y.Y. and F.L.; investigation, Y.Z., J.Z., Y.Y., L.L. and W.K.; resources, W.K., J.Z., F.L. and Y.Y.; data curation, W.K., J.Z. and F.L.; writing—original draft prep-aration, Y.Z.; writing—review and editing, W.K., Y.Z. and F.L.; visualization, Y.Z., J.Z., Y.Y. and L.L.; project administration, W.K. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Science and Technology Department of Zhejiang Province (2021C02023), Science and Technology Department of Shenzhen (CJGJZD20210408092401004), and Zhejiang Provincial Education Department Scientific Research Project (Y202147218).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to it belongs to the orchard base.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Liu, B. *Research on Crop Classification Based on UAV Remote Sensing Images*; Chinese Academy of Agricultural Sciences: Beijing, China, 2019.
2. Rasmussen, J.; Ntakos, G.; Nielsen, J.; Svensgaard, J.; Poulsen, R.N.; Christensen, S. Are vegetation indices derived from consumer-grade cameras mounted on UAVs sufficiently reliable for assessing experimental plots? *Eur. J. Agron.* **2016**, *74*, 75–92. [CrossRef]
3. Tan, L.; Lv, X.; Lian, X.; Wang, G. YOLOv4_Drone: UAV image target detection based on an improved YOLOv4 algorithm. *Comput. Electr. Eng.* **2021**, *93*, 107261. [CrossRef]
4. Kou, M.; Zhuo, L.; Zhang, J.; Zhang, H. Research advances on object detection in Unmanned Aerial Vehicle imagery. *Meas. Control. Technol.* **2020**, *39*, 47–61. [CrossRef]
5. Deng, J.; Ren, G.; Lan, Y.; Huang, H.; Zhang, Y. UAV ultra-low altitude remote sensing image processing based on visible light band. *J. South China Agric. Univ.* **2016**, *37*, 16–22.
6. Fang, Y.; Qiu, X.; Guo, T.; Wang, Y.; Cheng, T.; Zhu, Y.; Chen, Q.; Cao, W.; Yao, X.; Niu, Q.; et al. An automatic method for counting wheat tiller number in the field with terrestrial LiDAR. *Plant Methods* **2020**, *16*, 132. [CrossRef]
7. Fernandez-Gallego, J.A.; Buchaillot, M.L.; Aparicio Gutiérrez, N.; Nieto-Taladriz, M.T.; Araus, J.L.; Kefauver, S.C. Automatic wheat ear counting using thermal imagery. *Remote Sens.* **2019**, *11*, 751. [CrossRef]
8. Zhou, C.; Liang, D.; Yang, X.; Xu, B.; Yang, G. Recognition of wheat spike from field based phenotype platform using multi-sensor fusion and improved maximum entropy segmentation algorithms. *Remote Sens.* **2018**, *10*, 246. [CrossRef]
9. Li, Y.; Du, S.; Yao, M.; Yi, Y.; Yang, J.; Ding, Q.; He, R. Method for wheatear counting and yield predicting based on image of wheatear population in field. *Nongye Gongcheng Xuebao Trans. Chin. Soc. Agric. Eng.* **2018**, *34*, 185–194.
10. Shrestha, B.L.; Kang, Y.M.; Yu, D.; Baik, O.D. A two-camera machine vision approach to separating and identifying laboratory sprouted wheat kernels. *Biosyst. Eng.* **2016**, *147*, 265–273. [CrossRef]
11. Cao, M.; Zhang, L.; Wang, Q. Rapid extraction of street tree information from UAV remote sensing images. *J. Cent. South Univ. For. Sci. Technol.* **2016**, *36*, 89–93.
12. He, Y.; Zhou, X.; Huang, H.; Xue, Q. Extraction of subtropical forest stand numbers based on UAV remote sensing. *Remote Sens. Technol. Appl.* **2018**, *33*, 168–176.
13. Teng, W.; Wen, X.; Wang, N.; Shi, H. Single-wood canopy extraction from high-resolution remote sensing images based on iterative H-minima improved watershed algorithm. *Adv. Lasers Optoelectron.* **2018**, *55*, 499–507.
14. Narkhede, P.R.; Gokhale, A.V. Color image segmentation using edge detection and seeded region growing approach for CIELab and HSV color spaces. In Proceedings of the 2015 International Conference on Industrial Instrumentation and Control (ICIC), Pune, India, 28–30 May 2015; pp. 1214–1218.
15. Jermsittiparsert, K.; Abdurrahman, A.; Siriattakul, P.; Sundeeva, L.A.; Hashim, W.; Rahim, R.; Maseleno, A. Pattern recognition and features selection for speech emotion recognition model using deep learning. *Int. J. Speech Technol.* **2020**, *23*, 799–806. [CrossRef]
16. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Las Vegas, NV, USA, 27–30 June 2016.
17. Ren, S.; He, K.; Girshick, R. Faster R-CNN: Towards real-time object detection with region proposal net-works. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
18. Dai, J.; Li, Y.; He, K. R-fcn: Object detection via region-based fully convolutional networks. *Adv. Neural Inf. Processing Syst.* **2016**, *29*.
19. He, K.; Gkioxari, G.; Dollár, P. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
20. Redmon, J.; Divvala, S.; Girshick, R. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
21. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
22. Chen, R.C. Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning. *Image Vis. Comput.* **2019**, *87*, 47–56.
23. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.

24. Mahto, P.; Garg, P.; Seth, P.; Panda, J. Refining Yolov4 for vehicle detection. *Int. J. Adv. Res. Eng. Technol. (IJARET)* **2020**, *11*, 409–419.
25. Huang, Y.; Fang, L.; Huang, S.; Gao, H.; Yang, L.; Lou, X. Research on tree crown extraction based on improved Faster R-CNN model. *For. Resour. Manag.* **2021**, *1*, 173–179. [CrossRef]
26. Jing, W.; Hu, H.; Cheng, C.; Li, C.; Jing, X.; Guo, Z. Ground apple identification and counting based on deep learning. *Jiangsu Agric. Sci.* **2020**, *48*, 210–219.
27. Chen, F.; Zhu, X.; Zhou, W.; Gu, M.; Zhao, Y. Spruce counting method based on improved YOLOv3 model in UAV images. *J. Agric. Eng.* **2020**, *36*, 22–30.
28. Zheng, Y.; Wu, G. YOLOv4-Lite–Based Urban plantation tree detection and positioning with high-resolution remote sensing imagery. *Front. Environ. Sci.* **2022**, *9*, 756227. [CrossRef]
29. Yang, B.; Gao, Z.; Gao, Y.; Zhu, Y. Rapid detection and counting of wheat ears in the field using YOLOv4 with attention module. *Agronomy* **2021**, *11*, 1202. [CrossRef]
30. Wu, D.; Lv, S.; Jiang, M.; Song, H. 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.* **2020**, *178*, 105742. [CrossRef]
31. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual attention network for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3156–3164.
32. Yu, B.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. Bilateral segmentation network for real-time semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 325–341.
33. Ba, R.; Chen, C.; Yuan, J.; Song, W.; Lo, S. SmokeNet: Satellite smoke scene detection using convolutional neural network with spatial and channel-wise attention. *Remote Sens.* **2019**, *11*, 1702. [CrossRef]
34. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
35. Jaderberg, M.; Simonyan, K.; Zisserman, A. Spatial transformer networks. *Adv. Neural Inf. Processing Syst.* **2015**, *28*, 2017–2025.
36. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
37. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
38. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
39. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 1314–1324.
40. Liang, X.; Zhang, J.; Zhuo, L.; Li, Y.; Tian, Q. Small object detection in unmanned aerial vehicle images using feature fusion and scaling-based single shot detector with spatial context analysis. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 1758–1770. [CrossRef]
41. Huang, L.; Wang, Y.; Xu, Q.; Liu, Q. Recognition of abnormally discolored trees caused by pine wilt disease using YOLO algorithm and UAV images. *J. Agric. Eng.* **2021**, *37*, 197–203.
42. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
43. Yang, J.; Feng, Q.; Wang, S.; Zhang, J.; Yang, S. Method for detection of farmland dense small target based on improved YOLOv4. *J. Northeast. Agric. Univ.* **2022**, *53*, 69–79. [CrossRef]