Pitaya detection using an improved lightweight Faster R-CNN based on MobileNetV3 in densely planted pitaya orchards

Yulong Nan,[1] Huichun Zhang,[2,3] Jiaqiang Zheng,[2] Kunqi Yang[2]

[1]School of Mechanical Engineering, Yancheng Institute of Technology, Yancheng; [2]College of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing; [3]Co-Innovation Center of Efficient Processing and Utilization of Forest Resources, Nanjing Forestry University, Nanjing, China

Abstract

Accurate and rapid fruit detection was very important for robot picking precisely, so the large model size and slow detection speed of the detection algorithm are problems that need to be solved urgently. An improved lightweight Faster R-CNN based on MobileNetV3 was proposed in this paper, which was used to detect fruits on Ori_RGB and Rb_RGB image datasets that collected by RGB-D camera in densely planted commercial pitaya orchards. On the Rb_RGB image datasets, the detection AP of 0.929 and 0.898 were obtained using MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN, which decreased 1.38% and 4.67% than that using VGG16_FRCNN respectively, and the detection time was 35.4 and 18.8 ms per image, which decreased 46.5% and 71.6% than that using VGG16_FRCNN respectively. On the Ori_RGB image datasets, the detection AP of 0.911 and 0.856 were obtained using MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN, which decreased 2.15% and 8.06% than that using VGG16_FRCNN respectively, and the detection time was 35.2 and 19.5 ms per image, which decreased 47.2% and 70.8% than that using VGG16_FRCNN respectively. Weight sizes of MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN were 3.19%, 1.15% of that of VGG16_FRCNN respectively. The detection AP values on the Rb_RGB image test set using three networks than that on Ori_RGB image test set increased 1.98%, 4.91%, and 1.18%, but image type had no significant effect on AP. The improved lightweight Faster R-CNN based on MobilenetV3 is expected to deploy to the embedded system of the fruit picking robot to detect pitaya, which would promote the development of robot picking technology.

Correspondence: Huichun Zhan, College of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing, China. E-mail: njzhanghc@hotmail.com

Contributions: all the authors made a substantive contribution, read and approved the final version of the manuscript and agreed to be accountable for all aspects of the work.

Conflict of interest: the authors declare no competing interests, and all authors confirm accuracy.

Funding: this work is supported by National Natural Science Foundation of China (NSFC 32402424 and NSFC 32171790), National Key Research and Development Program of China (2023YFE0123600), Jiangsu Province Agricultural Science and Technology Independent Innovation Funds Project (CX(23)3126).

Acknowledgments: the authors would like to thank Ms. Ying Kaifang comes from the family farm of Sanzao Village, Xinxing Town, Tinghu District, Yancheng City for providing permission and venue for taking photos in the densely planted pitaya orchard.

Received: 19 June 2024.
Accepted: 30 June 2025

Journal of Agricultural Engineering 2025; LVI:1886
doi:10.4081/jae.2025.1886

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

Introduction

Pitaya has high nutritional value, rich in vitamin C and water-soluble dietary fiber, and is very popular with consumers (Jiang *et al*., 2021). In 2020, the area of pitaya planted in China was about 60,000 hectares, and the planting area would be expanding year by year. For most commercial orchards, fruit picking requires a large amount of labor. In the peak season for fruit harvesting, labor was facing a severe shortage. In addition, the cost of manual fruit picking accounted for 25% of the annual fruit production cost, and workers who picked fruits for a long time faced health risks (Li *et al*., 2019).

Robot picking could solve the huge risk of labor shortage to pick fruit in time, and it was an important way to replace manual picking (Li *et al*., 2022; Tang *et al*., 2020). Fruit picking robots were essential to alleviate labor shortages, reduce orchard production costs and the risk of labor injuries (Hou *et al*., 2021). A typical fruit-picking robot was mainly composed of a vision system and a mechanical actuator (Miao and Zheng, 2020). The vision system was used to detect and locate the fruit position (Wang *et al*., 2022), and guided the mechanical actuator to pick the fruit precisely (Chiu *et al*., 2013; Mu *et al*., 2020).

Fruit detection was a necessary first step for robots to realize automatic picking, and image technology was one of the main fruit detection methods (Ni *et al*., 2018). Traditional image processing used features such as color (Malik *et al*., 2018), shape (Bargoti and Underwood, 2017), and texture (Sengupta and Lee,

2014) to detect fruits, and the detection results were severely interfered with by factors such as variable lighting conditions, fruit clusters occlusion, and complex backgrounds (Vasconez *et al*., 2020). A variety of improved deep learning networks were used to detect fruits such as citrus (Yang *et al*., 2019), kiwi (Suo *et al*., 2021; Wan and Goudos, 2020; Williams *et al*., 2019), apple (Wan and Goudos, 2020), sweet pepper (Arad *et al*., 2020) and lychee (Yu *et al*., 2021), and the detection accuracy was 85.3~91.5%. Faster Region-based Convolutional Neural Network (Faster R-CNN or FRCNN) has been used for the intelligent detection of many different fruits (Fu *et al*., 2020a. Multi-class apples-on-plant in the SNAP system were detected using Faster R-CNN with ZFNet and VGG16 as the backbone network, and the mean average precision (mAP) of apple detection was 87.9%, and the apple detection speed was 0.241s per image on average (Gao *et al*., 2020). The RGB-D camera (Kinect v2) was used to filter the image background using depth features, which improved the average precision (AP) of apple detection by 2.5% using Faster R-CNN (Fu *et al*., 2020b). Multi-modal images (color, depth, and intensity) were used by adapted Faster R-CNN, and detection results showed an improvement of 4.46% in F1-score when adding depth and intensity channels of apple images (Gené-Mola *et al*., 2019). The Faster R-CNN was improved by optimizing the structure of the convolutional layer and the pooling layer in the network, and the results showed that the mAP of fruit detection was 90.7% (Wan and Goudos, 2020). Low and high features were extracted by multiple-scale feature extractors from the color and depth images collected by RGB-D camera, which was combined with Faster R-CNN to propose MS-FRCNN (multiple-scale Faster R-CNN) method to detect small passion fruit that increased F1-score from 0.885 to 0.946 (Tu *et al*., 2020). An improved yolov4-tiny model is proposed to detect Camellia oleifera fruit, and the detection AP was 0.921 (Tang *et al*., 2023). A lightweight convolutional neural network YOLOv4-LITE was proposed to detect pitaya, the detection AP was 96.5%, and the detection time of a single 1200×900 image was only 2.28 ms (Wang *et al*., 2020). However, the pitaya images collected in this study (Wang *et al*., 2020) were derived from web crawlers, and the detection model in the actual commercial orchard scene urgently needed further research and verification.

Therefore, an improved lightweight Faster R-CNN based on MobileNetV3 is proposed to detect pitaya in densely planted orchards in the paper. The main contributions of this work are as follows:

i) Pitaya image datasets from a densely planted commercial orchard were obtained;
ii) An improved lightweight Faster R-CNN based on MobileNetV3 was proposed to reduce the size of the detection model and improve the detection speed;
iii) The effect of the dataset type (Ori_RGB and Rb_RGB image datasets) on the detection of AP is verified.

Materials and Methods

Images acquisition

The complex environment of commercial orchards and changes in natural light are important obstacles for traditional image processing technology to achieve high-precision fruit detection. Using the depth distance information of the depth image from the RGB-D camera, not only the spatial position of the fruit can be estimated, but also most of the interfering background in the image of the orchard environment can be removed.

In this study, an image acquisition system established by Intel Realsense D435i depth camera (Intel Corporation, Santa Clara, CA, USA) was used to collect RGB and depth images in a densely planted commercial pitaya orchard. D435i depth camera uses the principle of stereo vision to obtain depth images. Compared with depth cameras using TOF technology, it is less sensitive to infrared radiation from sunlight and is suitable for indoor and outdoor environments. The working range of the D435i depth camera is 0.105~10m, and the depth measurement error within the 2 m working range indoor and outdoor is less than 2%.

This commercial pitaya orchard was located in Sanzao Village, Xinxing Town, Tinghu District, Yancheng City, Jiangsu Province, China, and the geographical location was 33°26′16.13″N and 120°5′8.12″E. In the commercial pitaya orchard, the row spacing of pitaya trees is 2m, the plant spacing is about 0.2 m, and the plant height is about 1.2 m. The fruits are mainly distributed on the branches of pitaya trees, as shown in Figure 1. The self-built image acquisition device was shown in Figure 1, which mainly includes a D435i depth camera, notebook computer, and tripod.

The D435i depth camera was installed on a tripod bracket through a universal platform, placed vertically and about 70 cm above the ground. This configuration allowed obtaining images of the entire pitaya trees in the height direction so that fruits at any height on the trees could be photographed. The computer was connected to the D435i depth camera through the USB3.1 data cable to drive the camera to work.

The field of view of the D435i camera depth sensor was 87°×58°, and the field of view of the RGB sensor was 69°×42°. Due to the difference in the field of view, it was challenging to register the two pieces of information at the same time. With the help of RealSense SDK2.0 and python software, set the configuration information of the D435i depth camera to align the depth image (1280x720) to the color image (1280x720), which made it more accurate and efficient to use the depth values of the depth image to remove the background from the RGB image. At the same time, relevant programs were written through python software to realize images collection and autosave.

The RGB image of a few pitaya trees obtained by using the D435i depth camera was shown in Figure 2a; and the depth image aligned with the RGB image was shown in Figure 2b. From the depth image and the distance unit of the depth pixel for the D435i

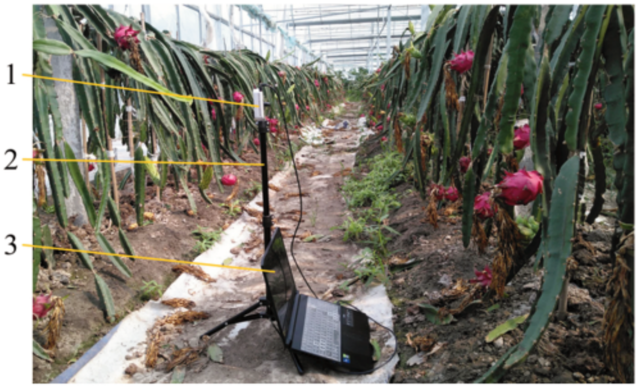

Figure 1. Pitaya trees in the densely planted commercial orchard and image acquisition device. 1, D435i depth camera; 2, tripod; 3, notebook computer.

depth camera, the depth distances corresponding to the pixel points of the RGB image could be obtained:

$$D_{RGB}(i,j)=imgD(i,j)\cdot I_d,\ (i=1,2,\cdots,H;\ j=1,2,\cdots,W) \qquad \text{(Eq. 1)}$$

where: D_{RGB} was the depth distances of the RGB image pixels from the D435i depth camera in the Z direction (mm); *imgD* was the depth image pixels; I_d was the distance unit of the depth image pixel value. In this experiment, the I_d of the D435i camera was 0.001mm/pixel. *i* was the pixel index number in the *H* direction of the image, and *j* was the pixel index number in the *W* direction of the image. *H* and *W* were the height and width of the image, respectively.

In the RGB image of the pitaya trees, The pitaya trees and their fruits in the target row were closer to the D435i camera than those in the non-target rows. Therefore, the depth distance threshold could be used to remove the background information of non-target rows, including pitaya trees, fruits, and other interference backgrounds in non-target rows. At the same time, the background information removed from the RGB image was filled with gray, as shown in Equation (2). In this paper, the depth distance threshold D_t was set to half of the row spacing (D_t = 1 m), and the pixels in the image area beyond D_t were set to gray to obtain an image with the background removed, as shown in Figure 2c.

$$\mathrm{Rb\_RGB}(i,j)=\begin{cases}\mathrm{Ori\_RGB}(i,j,1:3) & ,0<D_{RGB}\le D_t\\ gray\_color(1,1,1:3) & ,others\end{cases},\ (i=1,2,\cdots,H;\ j=1,2,\cdots,W) \qquad \text{(Eq. 2)}$$

where: Rb_RGB was the image after removing the background; Ori_RGB was the original RGB image; *gray_color* was the gray pixel value; D_t was the depth distance threshold (mm).

In the Rb-RGB images with the resolution of 720×1280 pixels, the ground truth pitaya targets were manually annotated using rectangular annotations through LabelImg software, as shown in Figure 3a, and then mapped to the corresponding Ori_RGB image (Fu *et al*., 2020b), as shown in Figure 3b.

A total of 4055 pitaya were labeled in the full set of pitaya Rb_RGB image datasets. The labeled pitaya Rb-RGB image datasets and the corresponding pitaya Ori-RGB image datasets were divided into a training set (710%, 910 images), validation set (15%, 195 images), and test set (15%, 195 images), respectively. The images in the training set were uniformly and randomly sampled from the entire datasets, and all images were not repeated to ensure the reliability of the later evaluation methods.

Improved lightweight Faster R-CNN based on MobileNetV3

Faster R-CNN (Ren *et al*., 2017) integrated feature extraction, region proposal network (RPN), bounding box regression, and classification into a network, which greatly improved the overall object detection performance, especially in terms of object detection speed. Faster R-CNN commonly used feature extraction backbone networks were ZFNet, VGG16, *etc*. (Gao *et al*., 2020).

MobileNetV3 was obtained by combining hardware-aware network architecture search (NAS) and NetAdapt algorithm to achieve novel network architecture improvements. MobileNetV3 was a lightweight network that combined 4 features such as deep separable convolution, inverse residual structure with linear bottleneck, squeeze-and-excitation networks, and the use of the H-swish function. Depth separable convolution was used to extract features, which was a combination of depthwise (DW) and pointwise (PW). MobileNetV3 contains two network versions that were MobileNetV3_large and MobileNetV3_samll, in which MobileNetV3_samll is 4 groups less inverse residual structure of linear bottleneck than that of MobileNetV3_large.

The MobileNetV3 network was divided into two parts: the feature extraction part and the classification part, to improve the Faster R-CNN model based on MobileNetV3. The feature extraction part included the ConvBNActivation (3x3) layer and X1 groups linear bottleneck inverse residual structure layer, where X1 equal to 1~12 for MobileNetV3_large; X1 equal to 1~8 for MobileNetV3_small. The classification part included X2 groups linear bottleneck inverse residual structure layer, ConvBNActivation (1x1), AdaptiveAvgPool2d, Reshape, Linear and H-swish layer, where X2 equal to13~15 for MobileNetV3_large, X2 equal to 9~11 for MobileNetV3_small. The feature extraction part and the classification part segmented from the MobileNetV3 network were used as the backbone and

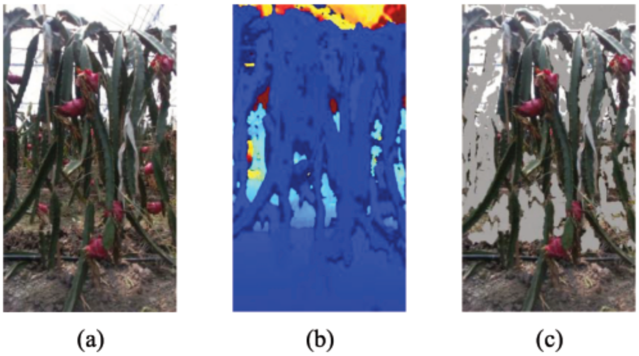

Figure 2. **a**) Original RGB image of a few pitaya trees; **b**) depth image aligned with RGB (pseudo-color); **c**) RGB image with background removed.

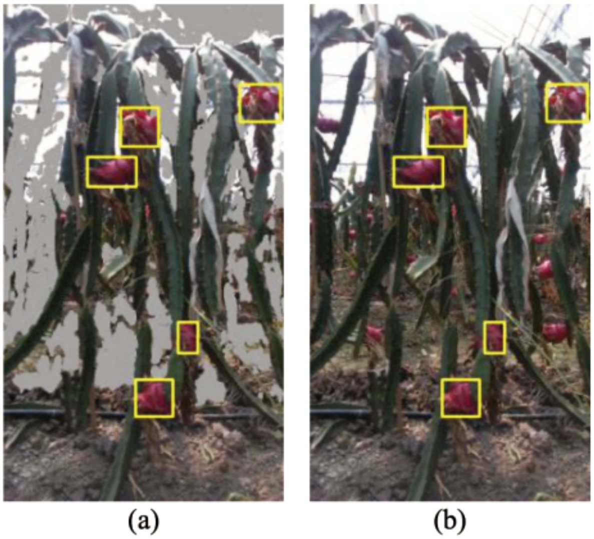

Figure 3. **a**) Use rectangular annotations to manually annotate the ground truth pitaya targets in the Rb-RGB image; **b**) the rectangular annotations were mapped to the Ori_RGB image.

classification parts of the Faster R-CNN, respectively, and an improved lightweight Faster R-CNN architecture based on MobileNetV3 was obtained, as shown in Figure 4.

Training procedures

The training platform was a computer equipped with an Intel Core i7-11700 (2.50 GHz) eight-core CPU, 16G memory, and an Nvidia GeForce GTX3060 GPU (12G video memory, 3584 CUDA core). The deep learning software environment was configured as Python 3.8, PyTorch 1.8.1, CUDA11.1, cuDNN8.0, and OpenCV 4.5. The improved lightweight Faster R-CNN framework based on MobileNetV3 could perform multi-class detection. This work only considered the binary classification problem of pitaya images acquired in the commercial orchard. Therefore, the output layer of the network was modified into two categories of background and pitaya regions, and the output layer was fully connected, and its modification did not affect the nonlinear mapping of high-order features (Abdalla *et al*., 2019). The Faster R-CNN based on VGG16 (Gao *et al*., 2020) was also used to train the pitaya detection network, which would be used to compare with pitaya detection performance using the improved lightweight Faster R-CNN based on MobileNetV3.

Transfer learning was used by network training, the adjusted input image shape was [800, 800, 3], the batch size was set to 2, and the training was performed for 300 epochs. The momentum of stochastic gradient descent was 0.95 and the weight decay parameter was set to 5e-4. The learning rate for the first 15 epochs was fixed to 1e-4, and the learning rate for the subsequent epochs was fixed to 1e-5. In this work, the shared convolution layers of the improved lightweight Faster R-CNN based on MobileNetV3 were initialized with the pre-trained ImageNet datasets classification network weight of MobileNetV3. The initial weights of the other layers of the network were initialized with a normal distribution with a standard deviation of 0.01 and a mean of zero. When comparing the predicted bounding box with the ground truth, the threshold of the Intersection over Union (IoU) was set to 0.5 to determine whether the detected instance was true (pitaya) or false.

The number of training images required by a deep learning network was affected by factors such as deep learning network architecture, image complexity, image enhancement technology, network learning parameters, and migration training methods. To determine the number of training images necessary when the deep learning network was trained with default parameters, a deep learning network training experiment in which the number of images in the training datasets gradually increased was performed (Fu *et al*., 2020b).

Performance evaluation

Use precision (P), recall (R), AP, and detection speed to evaluate the detection performance of the proposed network. Calculation methods of P and R were in eqs. (3) and (4), respectively (Yang *et al*., 2020).

$$P = \frac{TP}{TP+FP} \qquad \text{(Eq. 3)}$$

$$R = \frac{TP}{TP+FN} \qquad \text{(Eq. 4)}$$

where: TP, FP, and FN represent the number of correctly detected pitaya objects (true positives), the number of falsely detected pitaya objects (false positives), and the number of missing pitaya objects (false negatives), respectively. AP was the area under the P and R curve, and it was an evaluation index that measured the performance of the trained network, as shown in Equation (5).

$$AP = \int_0^1 P_{(R)} dR \qquad \text{(Eq. 5)}$$

Results

Performance evaluation

For the image datasets (Ori_RGB datasets and Rb_RGB datasets), randomly sample images from the training set of 910 pitaya images, and generate a subset of 10, 20, 50, 100, 300, 500, 700, and 910 pitaya images in sequence. As shown in Figure 5, the AP of MobileNetV3_large_FRCNN (improved lightweight Faster R-CNN based on MobileNetV3_large), MobileNetV3_small_FRCNN, and VGG16_FRCNN (Faster R-CNN based on VGG16) on the pitaya image test set vary with the number of training images. At the beginning of training with a small number of images (10~100 images), the AP values of the three training networks (MobileNetV3_large_FRCNN, MobileNetV3_small_FRCNN, and VGG16_FRCNN) on the image test set increased rapidly. When the number of training images was 10, the differ-

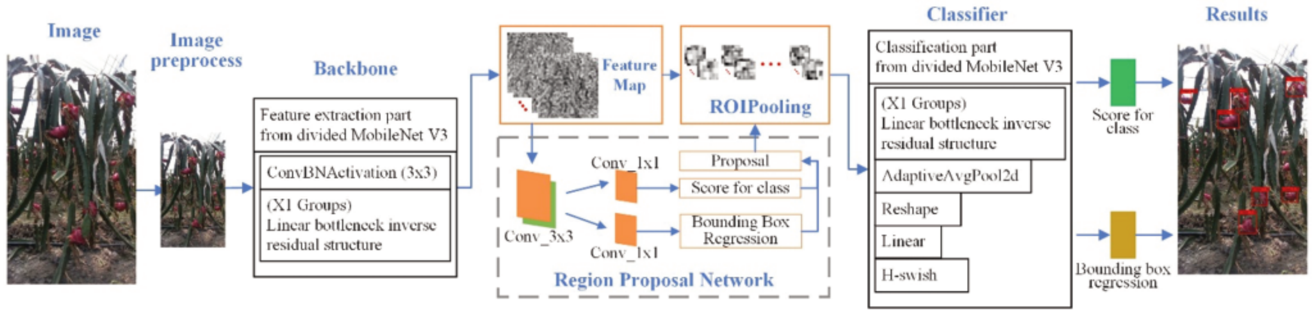

Figure 4. Improved lightweight Faster R-CNN based on MobileNetV3.

ence in AP values of the three training networks on the image test set was the largest. At this time, the AP of the VGG16_FRCNN network on the image test set was significantly better than that of MobileNetV3_large_FRCNN and MobileNetV3_small_FRCNN. With the gradual increase in the number of training pictures to 300, the difference in AP values of three training networks on the image test set decreased rapidly. After the number of training pictures reached 300, the AP values of three training networks on the image test set tended to converge.

A one-way analysis of variance (Nan *et al*., 2023) was used to test the significance for the number of training images to the AP by python software and pandas library, to determine the necessary number of training images for the three training networks to train with the default parameters. The results of Significance Level in Fig. 6 showed that the factor of the number of training images had no significant effect on AP at the significance level of 5% when the number of training images reached 300. Therefore, it was recommended that the number of training pictures needs to be greater than or equal to 300 to ensure the stability of the detection performance of the target detection network.

A one-way analysis of variance was used to test the significance for the factor of image type (Rb_RGB or Ori_RGB) to the AP by python software and pandas library, to verify whether the image type has a significant impact on AP. The results of significance level in Table 1 showed that the factor of image type had no significant effect on AP at the significance level of 5% for three deep learning object detection networks. Therefore, this image type (the images with background removed based on depth distance) has a certain positive effect on the improvement for the AP values of the three deep learning networks described in this paper, but the effect on the improvement of AP was not significant.

The Precision-Recall (P-R) curves were obtained by detecting the pitaya fruits on the Ori_RGB and Rb_RGB image test sets using the three object detection networks respectively, as shown in Figure 7. At the same R-value, the P values of the three object detection networks on the Rb_RGB image test set were higher than those on the Ori_RGB image test set. At the same R-value, the descending order of P-value obtained by three object detection networks to detect pitaya on two image types of test sets was:

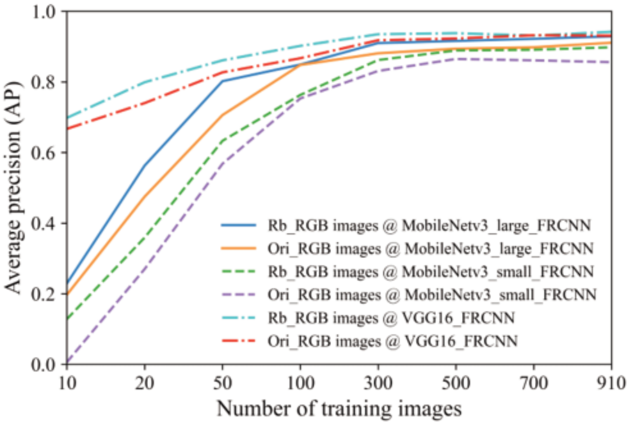

Figure 5. Average precision (AP) values of MobileNetV3_large_FRCNN, MobileNetV3_small_FRCNN, and VGG16_FRCNN on the pitaya image test set vary with the number of training images.

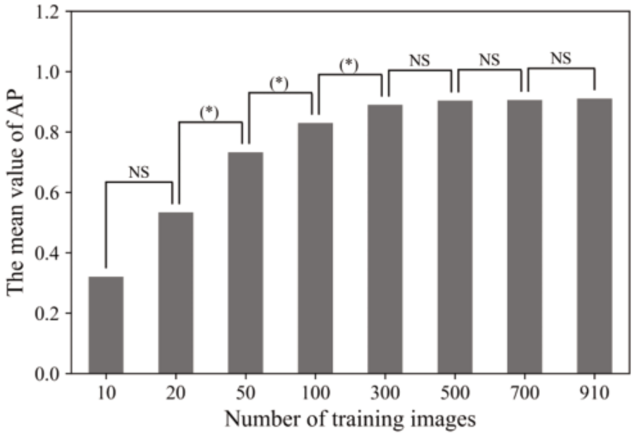

Figure 6. The results of significance level for the factor of the number of training images to the AP. *p<0.1; NS, not significant at p>0.1.

Table 1. One-way analysis of variance for the factor of image type to AP.

Object detection network	Item	df	Sum of squares	Mean square	F	Significance level
MobileNetV3_large_FRCNN	Factor (image type)	1	0.006	0.006	0.093	NS
	Residual	14	0.906	0.065		
MobileNetV3_small_FRCNN	Factor (image type)	1	0.011	0.011	0.113	NS
	Residual	14	1.324	0.095		
VGG16_FRCNN	Factor (image type)	1	0.003	0.003	0.288	NS
	Residual	14	0.123	0.009		

NS, not significant at p>0.05.

Table 2. The performance index results of pitaya detection on two image types of test sets using three networks.

Index	Image types	Networks		
		MobileNetv3_large_FRCNN	MobileNetv3_small_FRCNN	VGG16_FRCNN
AP	Rb_RGB	0.929	0.898	0.942
	Ori_RGB	0.911	0.856	0.931
Mean detection speed* (ms/image)	Rb_RGB	35.4^{b}	18.8^{c}	66.2^{a}
	Ori_RGB	35.2^{b}	19.5^{c}	66.7^{a}

*Mean detection speed may vary across different hardware settings and input image shape, the adjusted input image shape was [600,600,3] in predict process; different letters after each column of values indicate significant differences (p<0.05).

VGG16_FRCNN, MobileNetV3_large_FRCNN, MobileNetV3_small_FRCNN.

Comparison of pitaya detection performance

The performance index results of pitaya detection on two image types of test sets using three networks were shown in Table 2. From the perspective for AP value and detection speed of pitaya detection, the detection performances of the three networks on the two types of image datasets and the impact of the image type on the detection performance were compared as follows.

AP value of pitaya detection

The detection AP values on Rb_RGB image test set using MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN than that using VGG16_FRCNN decrease 1.38% and 4.67% respectively. The detection AP values on pitaya Ori_RGB image test set using MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN than that using VGG16_FRCNN decrease 2.15% and 8.06%, respectively. The detection AP values on the Rb_RGB image test set using three networks than that on Ori_RGB image test set increased 1.98%, 4.91%, and 1.18%. However, the results of significance level in Table 3 showed the factor of image type had no significant effect on AP at the significance level of 5%. Compared with using the VGG16_FRCNN network, The detection AP values on Rb_RGB and Ori_RGB image test sets using MobileNetv3_large_FRCNN and MobileNetv3_ small_FRCNN both showed a slight decrease. The reason was that depthwise separable convolutions were used in MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN, resulting in a small amount of feature information loss.

The detection AP values on Rb_RGB and Ori_RGB image test sets using MobileNetv3_small_FRCNN were 0.898 and 0.856 respectively, which decreased 3.34% and 6.04% than that using MobileNetv3_large_FRCNN respectively. The reason was that MobileNetv3_small_FRCNN simplified and removed 4 groups linear bottleneck inverse residual structure layer than MobileNetv3_large_FRCNN, which weakened the performance of image feature extraction to cause the AP value of pitaya detection using MobileNetv3_small_FRCNN to slightly decrease.

Detection speed of pitaya detection

Mean detection speeds of pitaya detection on Rb_RGB image test set using MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN were 35.4 and 18.8 ms/image, then the detection time decreased 46.5% and 71.6% than that using VGG16_FRCNN, respectively. The detection speeds on the Ori_RGB image test set using MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN were 32.5 and 19.5 ms/image, then the detection time decreased 47.2% and 70.8% than that using VGG16_FRCNN respectively. Multiple comparisons were used to analyze the significant difference in detection speed between any two groups' detection networks. The results of multiple comparison analysis showed that networks type had a significant difference in detection speed at the significance level of 5%, as shown in Table 2. This showed that the MobileNetv3_small_FRCNN network had significant advantages in detection speed.

Compared with using the VGG16_FRCNN network, the mean detection speed on Rb_RGB and Ori_RGB image test sets using MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN were both greatly improved. The reason was that depthwise separable convolutions were used in MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN, which greatly cut down the weight sizes of the networks to reduce the amount of calculation and improve detection speed. Weight sizes of MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN were 49.8 and 17.9MB respectively, which was 3.19%, 1.15% of that of VGG16_FRCNN respectively, as shown in Table 4.

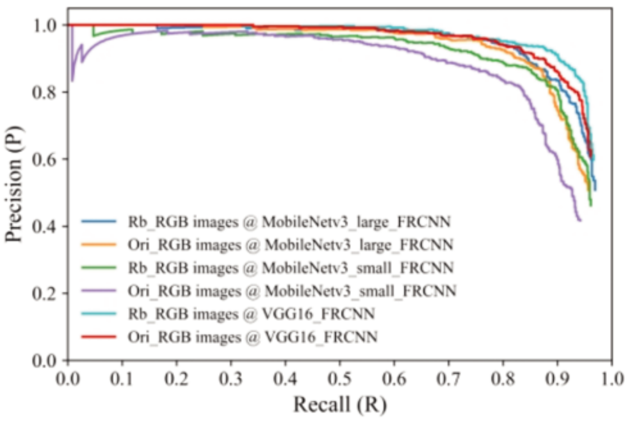

Figure 7. The Precision-Recall (P-R) curves obtained by detecting the pitaya fruits in the Ori_RGB and Rb_RGB pitaya image test sets using the three object detection networks, respectively.

Table 3. One-way analysis of variance for the factor of image type to AP.

Item	df	Sum of squares	Mean square	F	Significance level
Factor (image type)	1	8.40E-04	8.40E-04	0.832	NS
Residual	4	4.04E-03	1.01E-03		

NS, not significant at p>0.05.

Table 4. The weight size of three pitaya detection networks on two image types of test sets.

Item	Image types	Networks		
		MobileNetv3_large_FRCNN	MobileNetv3_small_FRCNN	VGG16_FRCNN
Weight size* (MB)	Rb_RGB Ori_RGB	49.8	17.9	1.56E+03

*Weight size may vary across input image shape, the adjusted input image shape was [800,800,3] in the training process.

The mean detection speed of pitaya detection on Rb_RGB and Ori_RGB image test sets using MobileNetv3_small_FRCNN was 18.8 and 19.5ms respectively, which decreased 46.89% and 44.46% than that using MobileNetv3_large_ FRCNN respectively. The reason was that MobileNetv3_small_FRCNN removed 4 groups linear bottleneck inverse residual structure layer than MobileNetv3_large_FRCNN, which further cut down the weight sizes of the network to reduce the amount of calculation and shorten the detection time. The weight size of MobileNetv3_ small_FRCNN was 17.9MB, which was 35.9% of that of MobileNetv3_large_FRCNN, as shown in Table 4.

Visual assessment

Visual assessment of three detection networks on a set of Rb_RGB and Ori_RGB images with the same number was shown in Figure 8. The yellow boxes represent the ground truth positions, the green boxes represent the detection position, and the differences among different detections using three networks on two types of images were manually marked with red circles in Figure 8. On this Ori_RGB image, all three networks could correctly detect fruits in the ground truth positions of the target row, but both MobileNetV3_small_FRCNN and VGG16_FRCNN incorrectly detect fruits in the non-target rows. On this Rb_RGB image, all three networks can correctly detect the dragon fruit at the ground truth positions of the target row. Since the background of the pitaya

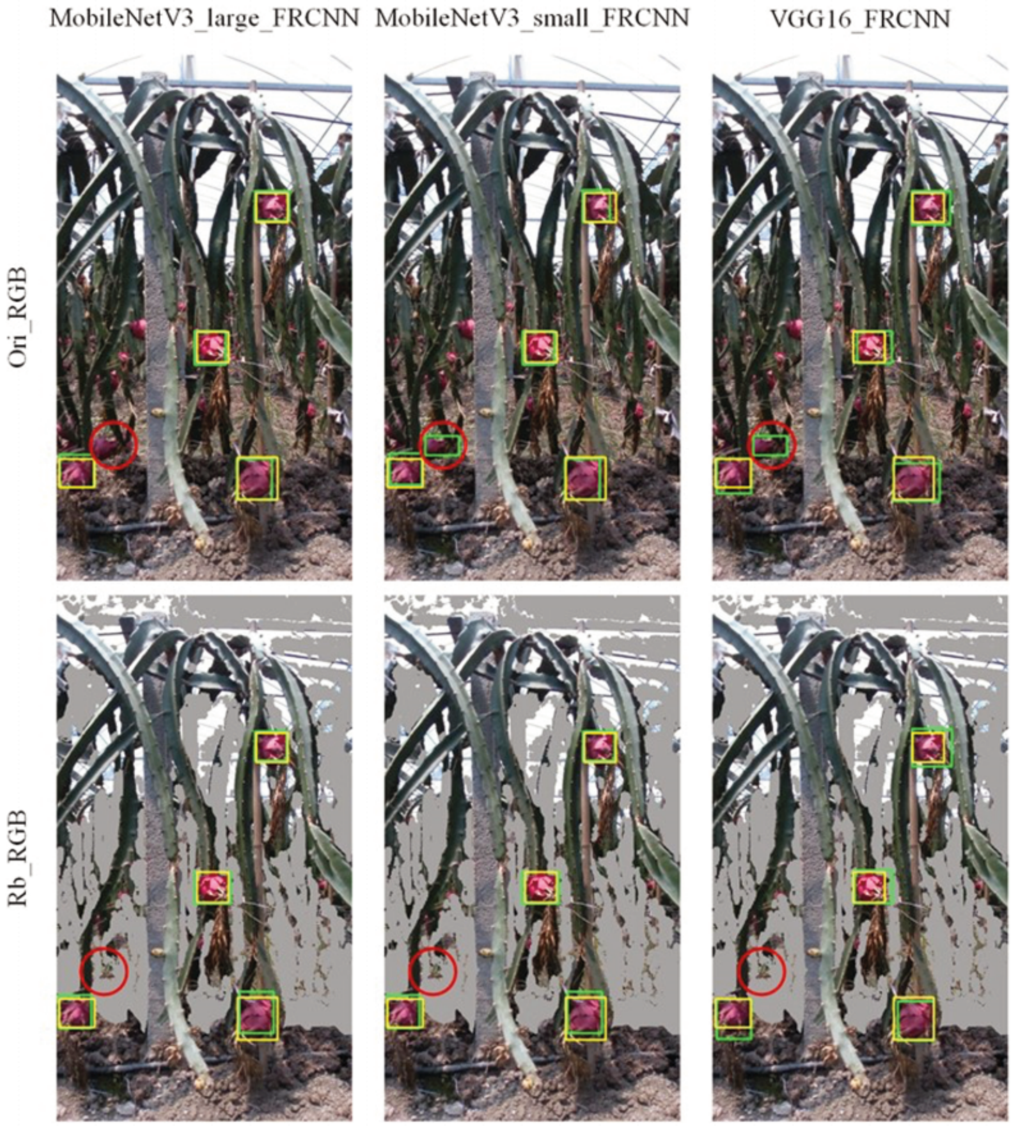

Figure 8. Visual assessment of three detection networks on a set of Rb_RGB and Ori_RGB images with the same number. The yellow boxes represent the ground truth positions; the green boxes represent the detection position; the differences among different detections using three networks on two types of images were manually marked with red circles.

trees and their fruits in the non-target rows were removed, there would be no false detection in the non-target rows. This was the reason why the AP values of the three networks on the Rb_RGB image test set were higher than those on the Rb_RGB image test set.

Discussion

Comparison with other fruit detection studies

The results of other fruit detection studies using Ori_RGB images were shown in Table 5. Wang *et al*. (2020) reported that the AP of pitaya detection was 96.5%, the detection speed was 2.28 ms per image, which was detected by using the YOLOv4-LITE network. In this research, The detection AP values using MobileNetv3_large_FRCNN or MobileNet_v3_samll_FRCNN network were lower than that reported by Wang *et al*. (2020). This was mainly because the pitaya image data set used in this study came from a densely planted commercial orchard, where densely planted pitaya trees and their fruits in non-target rows made the background environment of the target row pitaya image more complicated and interferential. However, the image data set came from a web crawler in the study of Wang *et al*. (2020), and the images had almost no background from pitaya trees and their fruits in the non-target rows. At the same time, the detection speed in this study was lower than that reported by Wang *et al*. (2020). This was mainly due to the difference in the resized input image shape of the detection network, hardware configuration, and batch processing quantity. Adjusted input image shape of detection network was [416,416,3] in the study of Wang *et al*. (2020), which was [600,600,3] in this research. Meanwhile, Suo *et al*. (2021) reported that the AP of kiwifruit detection was 91.9%, and the detection speed was 25.5 ms per image when using the YOLOv4 network to detect multiple types of kiwifruit.

Fu *et al*. (2020b) reported that the AP of Scifresh apple detection was 87.1%, and detection speed was 124ms per image, which was detected by using VGG16_FRCNN. Gené-Mola *et al*. (2020) reported that the AP of Fuji Apple detection was 92.7%, and the detection speed was 74 ms per image, which was detected by using VGG16_FRCNN. In this research, the detection AP using MobileNetv3_large_FRCNN was slightly higher than that of Scifresh apple but lower than that of Fuji apple. At the same time, under the condition of a platform with a similar hardware configuration (number of GPU CUDA core), the detection time was shortened by 52.4% and 80.7% than that of Scifresh apple and that of Fuji apple.

The results of other fruit detection studies using Rb_RGB images were shown in Table 6. Fu *et al*. (2020b) reported that the AP of Scifresh apple detection was 87.1%, and detection speed was 181ms per image, which was detected by using Faster R-CNN based on VGG16 on Foreground-RGB images. In this research, the detection AP values using MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN were both slightly higher than that of Scifresh apple on Rb_RGB images. At the same time, under the condition of a platform with a similar hardware configuration (GPU Cuda core number), the detection time was shortened by 80.6% and 90.0% than that of Scifresh apple and that of Fuji apple on Rb_RGB images.

Potential applications and impact

The improved lightweight Faster R-CNN based on MobileNetv3 would be applied to pitaya detection during the process of robot picking fruits in densely planted commercial pitaya orchards and would provide position calculation parameters for the positioning of the robot picking end-effector in the further. The weights of MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN were 48.9M and 17.9M, respectively, which made the detection network lighter than that of VGG16_FRCNN, and greatly reduced the amount of calculation to increase the detection speed and cut down the power consumption of embedded devices. Therefore, the improved lightweight Faster R-CNN based on MobileNetv3 would be suitable for deployment on mobile embedded devices, which would meet the needs of embedded terminal use scenarios for actual commercial pitaya pickers.

Future work

At present, the AP of fire dragon fruit detected by MobileNetv3_large_FRCNN was 92.9%, and the detection AP still needed to be further improved. In the future, we would further improve the performance of the network from the following possible aspects: i) adding an improved attention mechanism network to the network; ii) using a weighted bi-directional feature pyramid network to improve the Head part of the existing network; and iii)

Table 5. The results of other fruit detection studies using Ori_RGB images.

Source	Fruit type	Image resolution	Main methods	Detection rate (%)	Detection speed (ms/image)
Gené-Mola *et al*. (2020)	Fuji Apple	548 × 373	Faster RCNN (VGG16)	92.7	74
Fu *et al*. (2020b)	Scifresh Apple	1920×1080	Faster RCNN (VGG16)	87.1	182
Suo *et al*. (2021)	kiwifruit	2352×1568	YOLOv4	91.9	25.5
Wang *et al* (2020)	Pitaya	416×416	YOLOv4-LITE	96.5	2.28
This research	Pitaya	1280×720	MobileNet_v3_large _FRCNN MobileNet_v3_samll _FRCNN	91.1 85.6	35.2 19.5

Table 6. The results of other fruit detection studies using Rb_RGB images.

Source	Fruit type	Image resolution	Main methods	Detection rate (%)	Detection speed (ms/image)
Fu *et al*. (2020b)	Scifresh Apple	1920×1080	Faster RCNN (VGG16)	89.3	181
This research	Pitaya	1280×720	MobileNet_v3_large _FRCNN MobileNet_v3_samll _FRCNN	92.9 89.9	35.4 18.8

improving the cross-entropy function of the existing network. The pitaya picking robot still faced the following problems in the actual picking process: i) the picking sequence of multiple contacting or shielding fruits; ii) the robot's picking and obstacle avoidance strategies when encountering the branches, supporting stone pillars, support rods, and steel wire ropes. Therefore, it was not enough to divide the detection categories into two categories (fruits and background) in the current detection. In future research, the detection categories will be further subdivided from the perspective of the robot picking strategies, and the picking robot would be used for picking verification in the densely planted commercial pitaya orchard.

Conclusions

An improved lightweight faster R-CNN based on MobileNetV3 was proposed in this paper, including MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN, which was used to detect fruits in densely planted commercial pitaya orchards with good performance.

On the Rb_RGB image datasets, the detection AP of 0.929 and 0.898 were obtained using MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN, which decreased 1.38% and 4.67% than that using VGG16_FRCNN respectively, and the detection time was 35.4 and 18.8 ms for per image, which decreased 46.5% and 71.6% than that using VGG16_FRCNN, respectively.

On the Ori_RGB image datasets, the detection AP of 0.911 and 0.856 were obtained using MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN, which decreased 2.15% and 8.06% than that using VGG16_FRCNN respectively, and the detection time was 35.2 and 19.5 ms for per image, which decreased 47.2% and 70.8% than that using VGG16_FRCNN, respectively.

Weight sizes of MobileNetv3_large_FRCNN and MobileNetv3_small_FRCNN were 49.8 and 17.9MB respectively, which was 3.19%, 1.15% of that of VGG16_FRCNN, respectively.

The detection AP values on the Rb_RGB image test set using three networks than that on Ori_RGB image test set increase 1.98%, 4.91%, and 1.18%, but image type has no significant effect on AP.

Therefore, the improved lightweight Faster R-CNN network based on MobileNetV3 not only obtained good detection AP but also greatly improved the detection speed. At the same time, the weight of the network was greatly reduced to cut down the amount of calculation, which was suitable for deployment to the embedded system of the pitaya picking robot.

Acknowledgments

The authors would like to thank Ms. Ying Kaifang comes from the family farm of Sanzao Village, Xinxing Town, Tinghu District, Yancheng City for providing permission and venue for taking photos in the densely planted pitaya orchard.

References

Abdalla, A., Cen, H., Wan, L., Rashid, R., Weng, H., Zhou, W., et al., 2019. Fine-tuning convolutional neural network with transfer learning for semantic segmentation of ground-level oilseed rape images in a field with high weed pressure. Comput. Electron. Agr. 167:105091.

Bargoti, S., Underwood, J.P., 2017. Image segmentation for fruit detection and yield estimation in apple orchards. J. Field Robot. 34:1039-1060.

Chiu, Y., Chen, S., Lin, J. 2013. Study of an autonomous fruit picking robot system in greenhouses. Eng. Agric. Environ. Food 6:92-98.

Fu, L., Gao, F., Wu, J., Li, R., Karkee, M., Zhang, Q., 2020a. Application of consumer RGB-D cameras for fruit detection and localization in field: A critical review. Comput. Electron. Agr. 177:105687.

Fu, L., Majeed, Y., Zhang, X., Karkee, M., Zhang, Q., 2020b. Faster R–CNN–based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting. Biosyst. Eng. 197:245-256.

Gao, F., Fu, L., Zhang, X., Majeed, Y., Li, R., Karkee, M., et al. 2020. Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN. Comput. Electron. Agr. 176:105634.

Gené-Mola, J., Sanz-Cortiella, R., Rosell-Polo, J. R., Morros, J., Ruiz-Hidalgo, J., Vilaplana, V., et al., 2020. Fruit detection and 3D location using instance segmentation neural networks and structure-from-motion photogrammetry. Comput. Electron. Agr. 169:105165.

Gené-Mola, J., Vilaplana, V., Rosell-Polo, J. R., Morros, J., Ruiz-Hidalgo, J., Gregorio, E., 2019. Multi-modal deep learning for Fuji apple detection using RGB-D cameras and their radiometric capabilities. Comput. Electron. Agr. 162:689-698.

Hou, Z., Li, Z., Fadiji, T., Fu, J., 2021. Soft grasping mechanism of human fingers for tomato-picking bionic robots. Comput. Electron. Agr. 182:106010.

Jiang, H., Zhang, W., Li, X., Shu, C., Jiang, W., Cao, J., 2021. Nutrition, phytochemical profile, bioactivities and applications in food industry of pitaya (Hylocereus spp.) peels: A comprehensive review. Trends Food Sci. Technol. 116:199-217.

Li, C. E., Tang, Y., Zou, X., Zhang, P., Lin, J., Lian, G., et al., 2022. A novel agricultural machinery intelligent design system based on integrating image processing and knowledge reasoning. Appl. Sci. 12:7900.

Li, Z., Miao, F., Yang, Z., Chai, P., Yang, S., 2019. Factors affecting human hand grasp type in tomato fruit-picking: A statistical investigation for ergonomic development of harvesting robot. Comput. Electron. Agr. 157:90-97.

Malik, M. H., Zhang, T., Li, H., Zhang, M., Shabbir, S., Saeed, A., 2018. Mature tomato fruit detection algorithm based on improved HSV and watershed algorithm. IFAC PapersOnLine 51:431-436.

Miao, Y., Zheng, J., 2020. Optimization design of compliant constant-force mechanism for apple picking actuator. Comput. Electron. Agr. 170:105232.

Mu, L., Cui, G., Liu, Y., Cui, Y., Fu, L., Gejima, Y., 2020. Design and simulation of an integrated end-effector for picking kiwifruit by robot. Inform. Process. Agr. 7:58-71.

Nan, Y., Zhang, H., Zheng, J., Yang, K., Ge, Y., 2023. Low-volume precision spray for plant pest control using profile variable rate spraying and ultrasonic detection. Front. Plant Sci. 13:1042769.

Ni, X., Wang, X., Wang, S., Wang, S., Yao, Z., Ma, Y., 2018. Structure design and image recognition research of a picking device on the apple picking robot. IFAC PapersOnLine 51:489-494.

Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE T. Pattern Anal. 39:1137-1149.

Sengupta, S., Lee, W.S., 2014. Identification and determination of the number of immature green citrus fruit in a canopy under different ambient light conditions. Biosyst. Eng. 117:51-61.

Suo, R., Gao, F., Zhou, Z., Fu, L., Song, Z., Dhupia, J., et al., 2021. Improved multi-classes kiwifruit detection in orchard to avoid collisions during robotic picking. Comput. Electron. Agr. 182:106052.

Tang, Y. C., Chen, M.Y., Wang, C.L., Luo, L.F., Li, J.H., Lian, G.P., et al. 2020. Recognition and localization methods for vision-based fruit picking robots: a review. Front. Plant Sci. 11:510.

Tang, Y., Zhou, H., Wang, H., Zhang, Y., 2023. Fruit detection and positioning technology for a Camellia oleifera C. Abel orchard based on improved YOLOv4-tiny model and binocular stereo vision. Expert Syst. Appl. 211:118573.

Tu, S., Pang, J., Liu, H., Zhuang, N., Chen, Y., Zheng, C., et al. 2020. Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images. Precis. Agric. 21:1072-1091.

Vasconez, J. P., Delpiano, J., Vougioukas, S., Auat Cheein, F., 2020. Comparison of convolutional neural networks in fruit detection and counting: A comprehensive evaluation. Comput. Electron. Agr. 173:105348.

Wan, S., Goudos, S., 2020. Faster R-CNN for multi-class fruit detection using a robotic vision system. Comput. Netw. 168:107036.

Wang, H., Zhao, Q., Li, H., Zhao, R., 2022. Polynomial-based smooth trajectory planning for fruit-picking robot manipulator. Inform. Process. Agric. 9:112-122.

Wang, J., Gao, K., Jiang, H., Zhou, H., 2020. Method for detecting dragon fruit based on improved lightweight convolutional neural network. T. CSAE 36:218-225.

Yang, C. H., Xiong, L. Y., Wang, Z., Wang, Y., Shi, G., Kuremot, T., et al., 2020. Integrated detection of citrus fruits and branches using a convolutional neural network. Comput. Electron. Agr. 174:105469.