Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 7351470, 12 pages
https://doi.org/10.1155/2021/7351470

Research Article

An Apple Detection Method Based on Des-YOLO v4 Algorithm for Harvesting Robots in Complex Environment

Wei Chen,[1,2] **Jingfeng Zhang**,[1] **Biyu Guo**,[1] **Qingyu Wei**,[1] **and Zhiyu Zhu**[1]

[1]*School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang 212100, China*
[2]*Jiangsu Maigao Information Technology Corporation, Zhenjiang 212002, China*

Correspondence should be addressed to Wei Chen; cw1@just.edu.cn

Received 21 May 2021; Revised 18 July 2021; Accepted 9 October 2021; Published 21 October 2021

Academic Editor: Yunchao Tang

Real-time detection of apples in natural environment is a necessary condition for robots to pick apples automatically, and it is also a key technique for orchard yield prediction and fine management. To make the harvesting robots detect apples quickly and accurately in complex environment, a Des-YOLO v4 algorithm and a detection method of apples are proposed. Compared with the current mainstream detection algorithms, YOLO v4 has better detection performance. However, the complex network structure of YOLO v4 will reduce the picking efficiency of the robot. Therefore, a Des-YOLO structure is proposed, which reduces network parameters and improves the detection speed of the algorithm. In the training phase, the imbalance of positive and negative samples will cause false detection of apples. To solve the above problem, a class loss function based on AP-Loss (Average Precision Loss) is proposed to improve the accuracy of apple recognition. Traditional YOLO algorithm uses NMS (Nonmaximum Suppression) method to filter the prediction boxes, but NMS cannot detect the adjacent apples when they overlap each other. Therefore, Soft-NMS is used instead of NMS to solve the problem of missing detection, so as to improve the generalization of the algorithm. The proposed algorithm is tested on the self-made apple image data set. The results show that Des-YOLO v4 network has ideal features with a mAP (mean Average Precision) of apple detection of 97.13%, a recall rate of 90%, and a detection speed of 51 f/s. Compared with traditional network models such as YOLO v4 and Faster R-CNN, the Des-YOLO v4 can meet the accuracy and speed requirements of apple detection at the same time. Finally, the self-designed apple-harvesting robot is used to carry out the harvesting experiment. The experiment shows that the harvesting time is 8.7 seconds and the successful harvesting rate of the robot is 92.9%. Therefore, the proposed apple detection method has the advantages of higher recognition accuracy and faster recognition speed. It can provide new solutions for apple-harvesting robots and new ideas for smart agriculture.

1. Introduction

The apple-harvesting robot is a comprehensive system that integrates environment perception, motion planning, and servo control. Among them, environmental perception is an important basis for harvesting robots to complete their picking tasks [1–3]. Robot systems usually use target detection technology to realize the function of environmental perception. Fast and accurate target detection can make robot work for a long time, reduce labor cost, and improve production efficiency [4–6]. Therefore, the research of apple detection has great significance to the improvement of the picking efficiency and success rate of the harvesting robot.

The recognition and positioning of fruits provide the target information for the robot control system. With the development of computer vision and artificial intelligence, there are more and more methods for target recognition and positioning [7–9]. Kelman et al. [10] realized the location of overlapping apples by analyzing multiple intensity profiles of fruit images. The accuracy of this method reaches 94%, but the calculation process takes a long time. Nyarko et al. [11] proposed a detection method of convex polyhedron approximation surface. This method has the advantages of simple calculation and efficient execution when the fruit is occluded. Wei et al. [12] proposed a fast segmentation method for color apple images. This method uses adaptive

mean shift and decision theory to determine the number of clusters and realizes the clustering segmentation of apple images. In order to solve the problem that it is difficult to process the apple images collected at night, Jai et al. [13] proposed a method combining differential image and color analysis to realize apple recognition at night. Song et al. [14] proposed an algorithm to detect and locate the fruiting branches of multiple litchi clusters in large environments. In this algorithm, DeepLabv3 is used to segment RGB image, and then nonparametric density space clustering method is used to cluster the pixels in the three-dimensional space of the tree skeleton image. The experimental results show that the detection accuracy of a litchi is 83.33% and the execution time of a single litchi is 0.464 s.

Due to the poor robustness of traditional vision methods in complex background, it is difficult to meet the work requirements of harvesting robots. In recent years, the CNN (convolutional neural network) [15–17] has been continuously improved, and it has shown great advantages in the field of target detection. It is mainly divided into two categories. The first type of CNN generates a series of target candidate boxes and then classifies the samples by convolutional neural network. Representative algorithms are R-CNN [18], Fast R-CNN [19], and Faster R-CNN [20]. Another kind of CNN directly transforms the problem of target border location into a regression problem, so it does not need to generate candidate boxes. The typical algorithms include SSD (Single Shot MultiBox Detector) [21] and YOLO (You Only Look Once) [22, 23]. Xu et al. [24] used machine learning methods to identify overlapping strawberries. Compared with the traditional segmentation method, this method can overcome the influence of light transformation. However, it is difficult to achieve good recognition results when the similarity between fruit and background is high. Wang et al. [25] proposed a method for identifying fruits and vegetables in an unstructured environment. The method used R-CNN model to identify fruits and vegetables and then completed the target location based on the principle of triangulation. Aiming at the problem that it is difficult to identify multicluster kiwi fruit in a complex field environment, Fu et al. [26] proposed a recognition method based on LeNet convolutional neural network. The recognition rate of this method for occluded fruit, overlapped fruit, adjacent fruit, and independent fruit was 78.97%, 83.11%, 91.01%, and 94.78%, respectively. However, the recognition rate of this method for partially occluded and overlapped fruit needed to be improved. Xiong et al. [27] used the Faster R-CNN detection model to detect green citrus in the natural environment. The experimental results showed that the comprehensive recognition rate of this method reached 77.45%, but the comprehensive recognition rate still needed to be further improved. Xue et al. [28] improved YOLO v2 to identify immature mangoes. The experimental results showed that the method can detect mangos at a speed of 83 f/s and an accuracy rate of 97.02%. However, from the perspective of recognition effect, the problem of missing recognition of fruits had yet to be solved. Inkyu et al. [29] used the ImageNet model to recognize sweet pepper, rock melon, apple, avocado, mango, and orange. The comprehensive recognition rate of this model reached 89.6%.

From the above analysis, it can be seen that it is difficult for conventional computer vision methods and deep learning methods to meet the technical requirements of harvesting robots. In order to make the harvesting robot recognize apples quickly and accurately in complex environment, traditional YOLO v4 algorithm is improved. Firstly, by drawing lessons from the DenseNet, the original structure of YOLO v4 is optimized to reduce model parameters effectively. This change can improve the ability of neural network to extract apple image features. Secondly, in order to solve the problem that the positive and negative samples of the collected data are not balanced in the training process, AP-Loss is used to improve the class loss function of YOLO v4. It can improve the accuracy of apple recognition. Finally, Soft-NMS replaces NMS to solve the problem of missing prediction boxes. It can improve the detection accuracy of apples under overlapping conditions. In order to verify the effectiveness of the Des-YOLO v4 algorithm, a harvesting experiment is carried out with the self-designed apple-harvesting robot.

2. Materials and Methods

2.1. Data Collection and Preprocessing. In this study, a variety of experimental materials in orchard and laboratory environments are collected for training and testing, so as to select the algorithm and parameters suitable for the apple-harvesting robot. The apple image was collected from the apple demonstration base in Dashahe Town, Jiangsu Province, China. The camera used in this study is a small camera OV2640, whose resolution is 1632×1232 pixels at 30 frames per second. It has small volume and low working voltage. Moreover, it can output sampling data of whole frame, subsampling, window, and so on. The camera is installed on the robot in eye-in-hand mode, so that the field of vision of the end effector and the camera does not interfere with each other in the process of fruit picking.

In order to reduce the probability of overfitting of the network model, the long-range and close-range images are collected. The distance from distant view and close view to fruit is 400–500 mm and 100–200 mm, respectively. In the case of distant view and close view, images from four directions of south, north, east, and west are collected, respectively, and two images are collected from each direction, with a total of 1600 images. To ensure the complexity of apple images, the image material should include the different numbers and occlusion of apples, as well as the lighting conditions such as natural light and backlight. As shown in Figure 1, it is a set of apple images in a typical complex environment. In the end, 2,000 image materials were collected, including captured images and 400 images of apples obtained by web crawlers, containing a total of 2,950 targets.

The training of YOLO neural network often requires more training sets. More training sets can make the neural network learn the features of apple image sufficiently and improve the generalization ability of the network model. However, in reality, due to the lack of material collection

Figure 1: Apple image in complex environment. (a) Single fruit. (b) Multiple fruits. (c) Occluded fruit. (d) Overlapping fruits. (e) Backlight. (f) Sunlight.

ability, it is difficult to obtain a large number of training materials. In addition, the growth posture of apples is different, and the overlap phenomenon is serious, so it is difficult to completely extract the shape characteristics of the fruit. Therefore, it is necessary to preprocess the apple image before the YOLO training. In this study, Matlab is used to process the original data set to achieve the effect of data enhancement.

(1) The image is rotated horizontally, vertically, or at a fixed angle, and the aspect ratio of the image is changed to generate more training sets

(2) Data are enhanced by adjusting saturation and hue, histogram equalization, median filtering, and other image processing techniques

(3) To improve the generalization ability of the model, four images are randomly cropped by Mosaic data enhancement method and spliced into one image as training data

After the image is processed by the above method, 10100 pictures are finally generated for later neural network training. LabelImg is used to mark the apple target in the above data set, and the marked information is saved in PASCAL VOC data set format. To ensure the uniform distribution of the data set, it is randomly divided into training set, verification set, and test set according to the proportion of 70%, 10%, and 20% by using Matlab tools. There are 7070 training set samples, 1010 verification set samples, and 2020 test set samples.

2.2. Apple Detection Based on YOLO v4. Apple detection is the information source of picking operations for harvesting robots, and it is also an important factor affecting the success rate of picking [30, 31]. This study uses the YOLO v4 algorithm to realize the recognition and positioning of apple targets, and it can locate the apples in a video and return their coordinates. YOLO v4 is one of the best detection algorithms at present. It has the advantages of fast recognition speed and high accuracy in apple detection. On the basis of the original YOLO v3 architecture, it introduces some optimization methods from data processing, backbone network, network training, activation function, loss function, and other aspects. YOLO v4 achieves the best matching in detection speed and accuracy so far [32–34].

The backbone network of YOLO v4 is CSPDarknet53, which is used to extract target features. YOLO v4 draws on the experience of the CSPNet (Cross Stage Partial Network) to maintain accuracy, reduce computing bottlenecks and memory costs, and add CSP to each large residual block of Darknet53 [35]. To reduce the amount of calculation and

ensure accuracy, YOLO v4 divides the feature mapping of the basic layer into two parts and then combines the hierarchical structure of different stages. The activation function of CSPDarknet53 uses the Mish function, and the rest of the network continues to use the Leaky_Relu function. Different from using FPN for upsampling in the YOLO v3 algorithm, YOLO v4 uses the idea of information flow in the PANet (Pyramid Attention Network) as a reference. The semantic information of high-level features is propagated to the low-level network through upsampling, and then it is combined with the high-resolution information of low-level features to improve the detection effect of small targets. As shown in Figure 2, the program flow of the YOLO v4 algorithm is as follows:

(1) The features of the input image are extracted through the backbone network, and then the input image is divided into $S * S$ grids ($S = 7$). If the center of a target is in a grid, this grid is responsible for the detection of the target.

(2) In order to complete the target detection, each grid needs to predict B bounding boxes and the categories probability of each bounding box and to output the confidence of whether the bounding box contains the target.

$$\begin{aligned} \mathrm{IOU} &= \frac{\mathrm{area}\left(\mathrm{box}\left(P\right) \cap \mathrm{box}\left(T\right)\right)}{\mathrm{area}\left(\mathrm{box}\left(P\right) \cup \mathrm{box}\left(T\right)\right)}, \\ \mathrm{confidence} &= \Pr\left(\mathrm{Object}\right) \times \mathrm{IOU}, \end{aligned} \tag{1}$$

where IOU (Intersection Union) is a standard performance measure between the predicted bounding box (box (P)) and the actual bounding box (box (T)). Pr (Object) is the probability that the current position has an object. If there is a target in the grid, Pr (Object) = 1; otherwise Pr (Object) = 0. Each bounding box contains five premeasurements: (x, y, w, h, confidence), where (x, y) represent the center coordinate values, (w, h) are the width and height of bounding box, and confidence is the confidence information.

(3) The category conditional probability C_i of each grid is calculated; then, the class-specific confidence score S_i of each bounding box can be obtained by multiplying the class conditional probability by the confidence of each bounding box.

$$\begin{aligned} C_i &= Pr\left(\mathrm{Class}_i|\mathrm{Object}\right), \\ S_i &= Pr\left(\mathrm{Class}_i|\mathrm{Object}\right) \times \Pr\left(\mathrm{Object}\right) \times \mathrm{IOU} \\ &= \Pr\left(\mathrm{Class}_i\right) \times \mathrm{IOU}. \end{aligned} \tag{2}$$

Pr (Class_i) is the category probability of the i-th target. By setting a threshold and comparing it with the S_i, the box whose score is lower than the threshold is filtered. Then, NMS is performed on the remaining boxes. Finally, the detection box of the target is obtained to realize the recognition and location of the apple.

This study obtains the two-dimensional coordinates (x_1, y_1) of apples through the detection box. The laser ranging sensor VL53L0 is used to measure the distance z between the target and the robot. Then, the three-dimensional coordinates (x, y, z) of the target in the camera coordinate system can be obtained by coordinate transformation formula (3). f is the focal length of the camera ($f = 3.6$ mm).

$$\begin{cases} x_1 = \dfrac{xf}{z}, \\ y_1 = \dfrac{yf}{z}. \end{cases} \tag{3}$$

2.3. Des-YOLO Network Structure Design. Because this study only detects the apples in the image, the structure of YOLO v4 network is optimized according to the DenseNet network. DenseNet enables feature information reuse through the connection layer by establishing the dense connection between the front layer and the back layer, thus reducing the amount of calculation. In DenseNet, all previous layers are connected as input:

$$x_l = H_l\left(\left[x_1, x_2, \cdots, x_{l-1}\right]\right), \tag{4}$$

where $[x_1, x_2, \ldots, x_{l\text{-}1}]$ is the mosaic of all feature maps before the layer. The above formula is a nonlinear mapping relationship. Because each layer receives the feature mapping from all the previous layers, the network can be thinner and more compact. Therefore, the number of channels can be reduced.

Based on the analysis and understanding of the network structure of DenseNet, a Des-YOLO network structure is proposed. The SPP (spatial pyramid pooling) block from the original YOLO v4 structure is removed, and a dense block is added in its position. Dense blocks can make the feature information be better transmitted in the whole network, and the situation of overfitting can be alleviated to some extent. YOLO v4 has three different sizes of anchors, which are 19, 38, and 76. In order to improve the detection speed, only 19×19 and 38×38 anchors are selected, because the larger the anchor is, the smaller the prediction box will be. If the prediction box is too small, the apple with a small resolution will be detected. In the process of picking, the distance between the apple with too small resolution and the manipulator is too far, so it is not the picking object in the current position. The structure of the Des-YOLO network is shown in Figure 3. The size of the input image is 416×416.

2.4. Optimization of Loss Function. The proponents of YOLO v4 believe that the design of loss function is one of the optimization techniques that can improve the accuracy without increasing the inference time. The prediction error of bounding box coordinates, the confidence error of

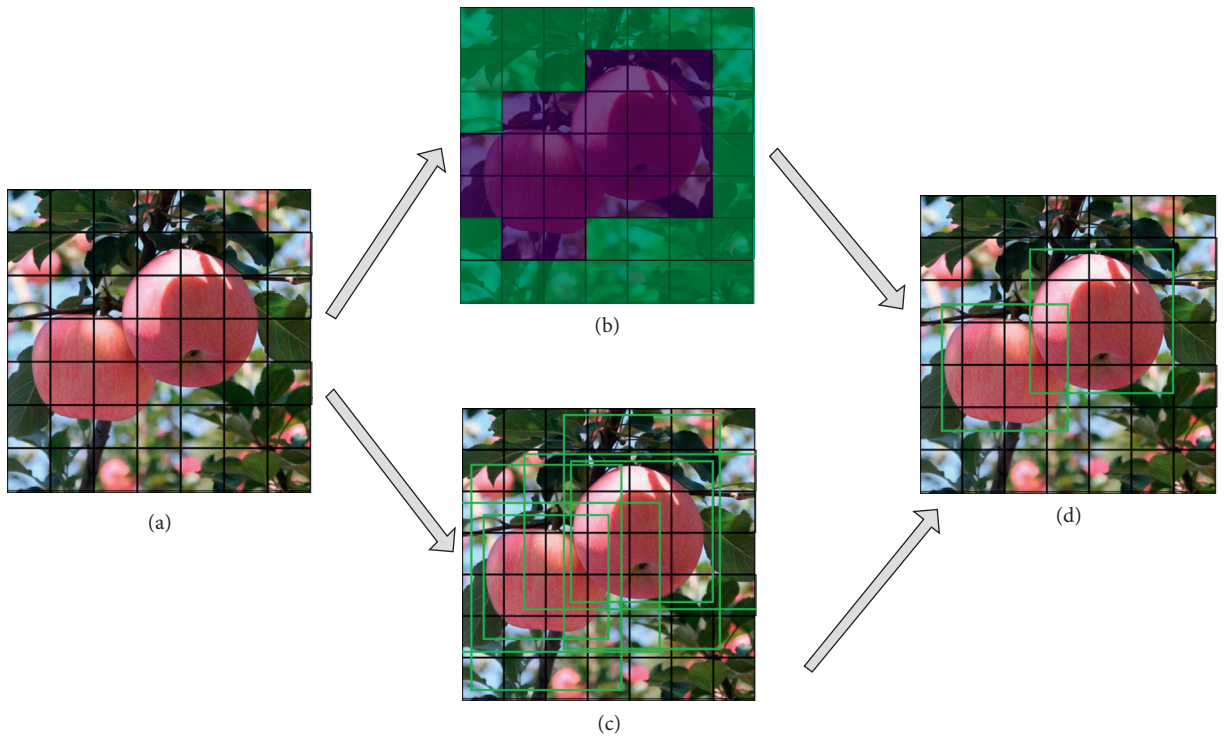


Figure 2: Apple detection based on YOLO v4. (a) Dividing image into $S * S$ grids. (b) Predicted class probability. (c) Regression bounding box. (d) Complete apple detection.

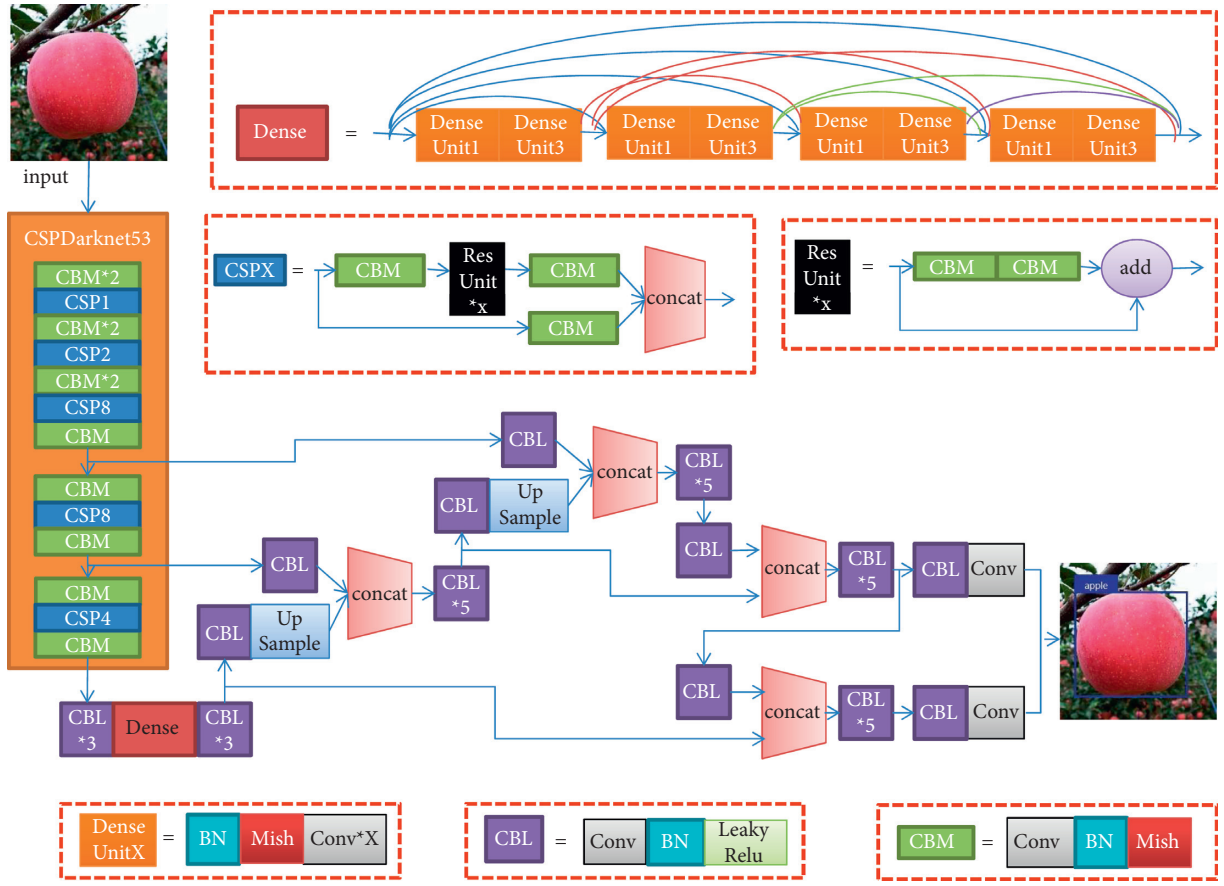


Figure 3: Network structure of Des-YOLO.

bounding box, and the prediction error of object category have been considered in the original loss function design.

YOLO v4 is a one-stage detection method. If the quantity gap between positive and negative samples is too large, it will reduce the accuracy of the network's recognition of apples. In order to solve the problem of imbalance between positive and negative samples, the category loss function based on AP-Loss (Average Precision Loss) is improved.

AP-Loss [36] transforms the classification task into the sorting task and minimizes the AP-Loss of the system based on the network error and its optimization algorithm. Firstly, the prediction box and score are transformed to obtain the transformation format of the prediction box and score, as shown in the following equations:

$$x_{km} = -(\alpha_k - \alpha_m), \tag{5}$$

$$y_{km} = 1|_{\beta_k=1,\beta_m=0}, \tag{6}$$

where K and M represent the k-th row and m-th column of an image, respectively; X_{KM} and Y_{KM} represent the difference of the overlap score of the two prediction frames and the converted score, respectively; and α and β represent the true value matching score and the original score of the anchor frame, respectively.

The network error is adjusted as follows:

$$L_{km}(x) = \frac{F(x_{km})}{1 + \sum_{n\in\Lambda\cup T, n\neq k} F(x_{kn})}, \tag{7}$$

where $F(x)$ is a sign function that, only if $x > 0$, takes 1; otherwise it is 0. $\Lambda$and $T$are the set of data groups marked with values 1 and 0, respectively.

The optimized loss function L_{cla} and its minimization objective function are shown in the following equations:

$$L_{cla} = \frac{|\Lambda| - 1}{|\Lambda|} \sum_{k\in\Lambda} \frac{1 + \sum_{m\in\Lambda, k\neq m} F(x_{km})}{1 + \sum_{m\in\Lambda, k\neq m} F(x_{km}) + \sum_{m\in T, k\neq m} F(x_{km})} = \frac{1}{|\Lambda|} \sum_{k\in\Lambda} \sum_{m\in T} L_{km} = \frac{1}{|\Lambda|} \langle L(x), y\rangle, \tag{8}$$

$$\min_{\delta} L_{cla}(\delta) = \frac{1}{|\Lambda|} \langle L(x, \delta), y\rangle, \tag{9}$$

where $\sum_{m\in\Lambda, k\neq m} F(x_{km})$ and $\sum_{m\in T, k\neq m} F(x_{km})$ are the ranking of α_k in positive samples and all valid samples, respectively. $L(x)$ and y are d-dimensional vectors composed of all L_{KM} and Y_{KM}, where d is the effective number of all prediction boxes and δ is the optimization parameter of the system.

The backpropagation gradient of the network is obtained by deriving the score function α_k, as shown in the following equation:

$$G_i = -\sum_{m,n} \Delta x_{mn} \frac{\partial x_{mn}}{\partial \alpha_k} = \sum_m L_{mk} y_{mk} - \sum_m L_{in} y_{km}. \tag{10}$$

2.5. Filtering Method of Prediction Box. In the test phase, the target detection algorithm will output multiple prediction boxes; in particular, there will be many high confidence prediction boxes around the target. In order to delete these duplicate prediction boxes and make each target have only one detection result, NMS (Nonmaximum Suppression) is generally used to filter the prediction boxes. Traditional NMS thinks that there is a clear boundary between targets. It will not produce too much overlap, so this algorithm can effectively remove false-positive samples and improve the detection accuracy. However, for the image containing multiple apples, the adjacent apples overlap with each other. According to the traditional NMS algorithm, some real apples with too high overlap will be directly removed from the detection queue, resulting in missed detection. In order to solve this problem, Soft-NMS [37] is used instead of NMS to filter prediction boxes.

Soft-NMS can make prediction boxes be revaluated recursively according to the current score, instead of being roughly deleted. In this way, it can avoid the situation of missing detection when multiple apples have a high overlap. At the same time, the algorithm does not need to retrain the model and does not increase the training cost. The algorithm flow is as follows:

(1) $B = \{B_1, \ldots, B_N\}$ is the prediction box set; $S = \{S_1, \ldots, S_N\}$ is the set of confidence scores corresponding to the prediction box

(2) $D = \{\}$ is the filtered prediction box set

(3) Select the box B_m with the highest score from set B, put it into set D, and assign the difference set of B and D to B

(4) If the IOU between the remaining box and B_m is greater than the set threshold N_T, the score will be reduced according to equation (11)

(5) Set the threshold N_d, and delete the box when the new score of the remaining box is less than N_d

(6) Repeat steps (3), (4), and (5) until B is an empty set, and then return D and S

For the prediction box with IOU greater than the threshold, a penalty function in the form of Gaussian function is constructed to reduce its score, as shown in the following equation:

$$s_i = s_i e^{-\left(iou\left(M, b_i\right)^2/\sigma\right)}, \tag{11}$$

where σ is the scale adjustment coefficient, given as 0.5 in this experiment. Soft-NMS changes the traditional method of directly removing the prediction box with high IOU and replaces it with the method of reducing its score. It reduces the probability of the correct prediction box being deleted by mistake and improves the average accuracy of detection.

3. Results and Discussion

3.1. Model Training and Detection Effect. In this experiment, the core processor of the training computer is AMD 3900 × 3.8 GHz CPU, and the graphics card is NVIDIA RTX

2080 Ti. The program is written by C++ and calls OpenCV, CUDA, and other operation libraries. In the aspect of model training, the learning rate is set to 0.001; momentum and decay are set to 0.9 and 0.0005, respectively; and the learning rate becomes 0.1 times the original after 11000 iterations.

After 12000 times of training, the loss function of the model changes as shown in Figure 4. It can be seen from the figure that in the first 1300 iterations, the loss function value decreases rapidly. The model is fitted rapidly and then gradually stabilizes after 3000 iterations. In the iterative process, the weight is output every 100 iterations, but the number of iterations is not the more the better. Too many iterations are prone to overfitting, so it is necessary to evaluate the model comprehensively.

The purpose of this study is to find suitable apples. Precision, Recall, mAP (mean Average Precision), and IOU are used to choose the appropriate threshold T ($0 < T < 1$) for the model. After the algorithm predicts the confidence of the target, the T needs to be compared with the confidence. The prediction targets with confidence higher than T are the apples that meet the harvesting requirements.

Figure 5 shows the change of mAP with the number of iterations. Among the models obtained in this experiment, models with higher mAP are selected, and then data experiments are carried out on these models. In this study, precision, recall, and IOU of these models are compared by constantly changing the threshold T, so that modes can detect the apples in the current environment according to needs.

In the apple recognition system, apples that are too far away or hidden behind the previous ones can be ignored because they will be recognized and located again before the next picking. Therefore, this study ignores the Recall and selects the Precision. For the IOU, because the harvesting robot only needs to recognize the center of apples, the requirements for the IOU are not high. To sum up, the priority of these parameters is Precision > Recall > IOU. The change of threshold T will change the Precision, Recall, and IOU of the detected target. When the threshold T is 0.5, the Precision and Recall are 97% and 90%, respectively, and the IOU is 83.61%. The performance of the model is at its best. The effect of the Des-YOLO v4 algorithm on the detection of apples in various environments in the test set is shown in Figure 6.

3.2. Experimental Comparison and Analysis. In order to further verify the efficiency of the improved model, the detection efficiency of various detection algorithms is compared. This study mainly evaluates the detection effects of YOLO v4, Faster R-CNN, and Des-YOLO v4 under the above conditions. In this experiment, multitarget images with different numbers and sizes are selected for detection experiment comparison, and the effect is shown in Figure 7. It can be seen that the Faster R-CNN detection efficiency is not high, and it is easy to miss the target. The conventional YOLO v4 algorithm has faster detection speed and detection accuracy, but there are many targets that are too far away in the detection results.

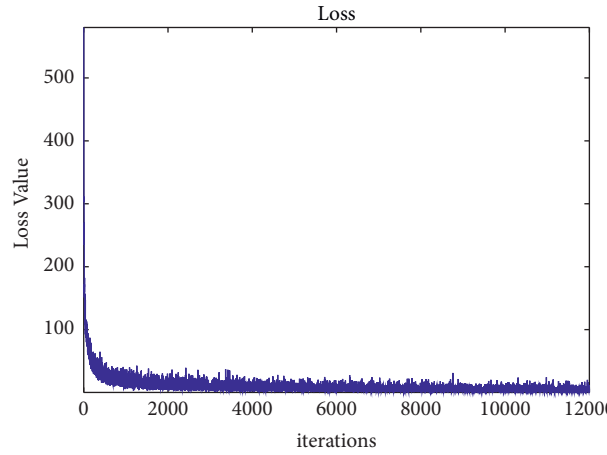


Figure 4: Loss function change chart.

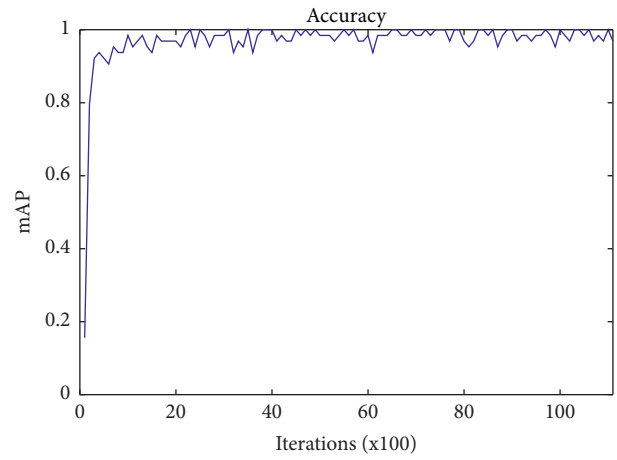
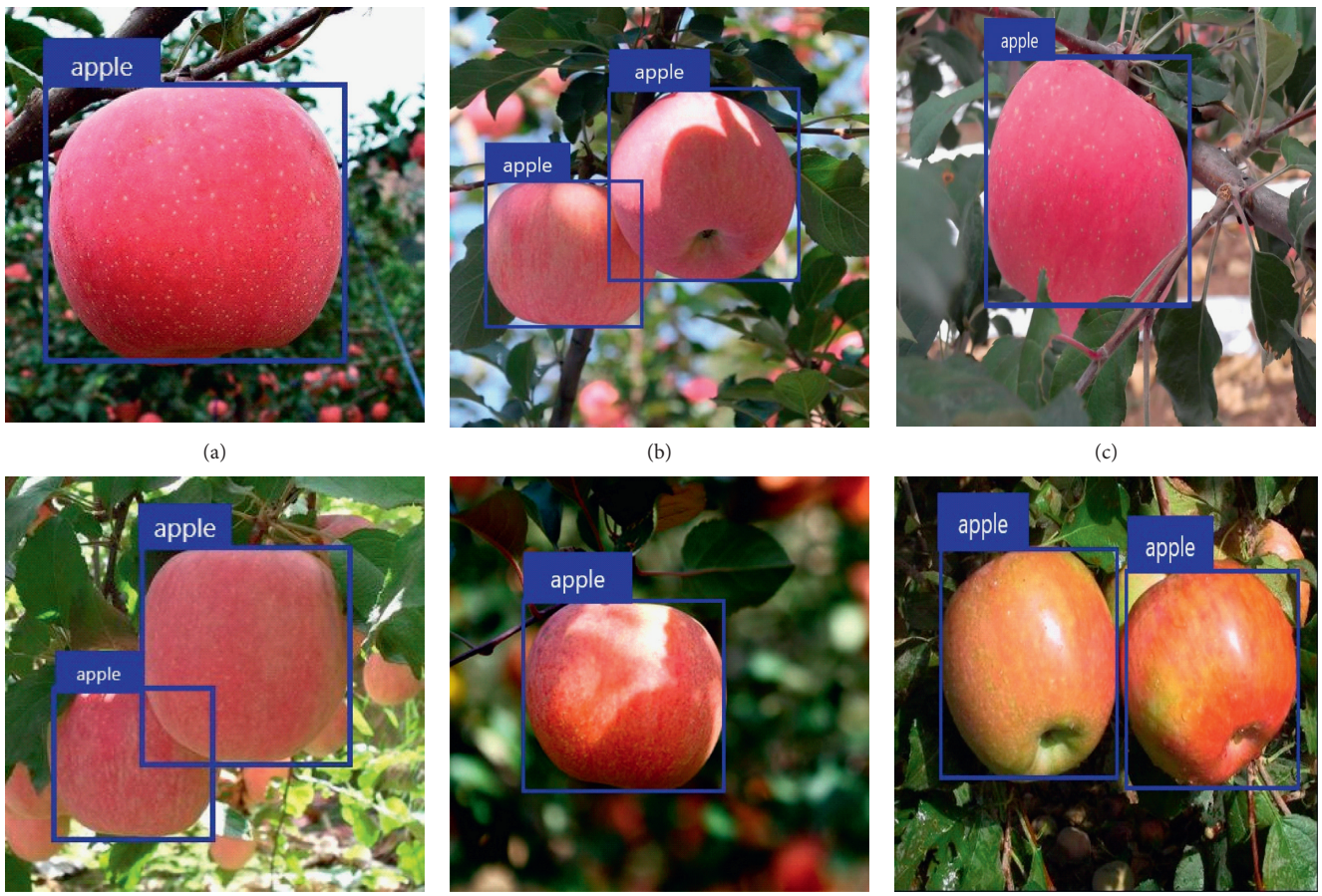


Figure 5: mAP change chart.

It can be seen from Table 1 that the Des-YOLO v4 algorithm performs better than the other algorithms in detecting apples. In the case of fewer apples, the detection results of several algorithms are similar, but the detection speed of the Des-YOLO v4 is faster and the mAP is relatively high. In the case of scattered apples, although Faster R-CNN can detect more apples, apple targets that are too far away cannot be picked in practical applications. In contrast, the Des-YOLO v4 algorithm has faster detection and higher detection accuracy. At the same time, the Des-YOLO v4 is better than the official YOLO v4 algorithm when there are more apple targets, so it is more suitable for harvesting robots. From the overall effect, the Des-YOLO v4 algorithm has a faster speed and a higher accuracy.

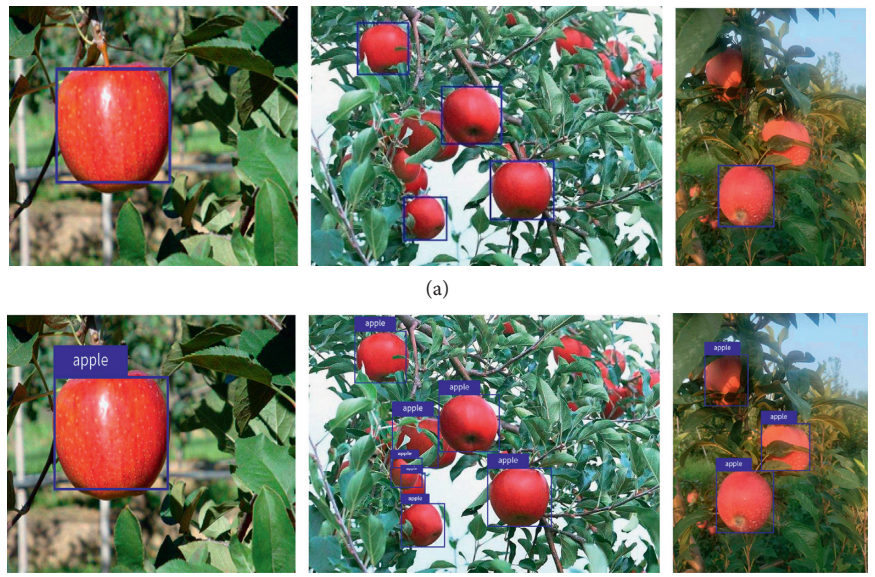
3.3. Robot Automatic Harvesting Experiment. The target detection and harvesting experiments are completed with a self-designed apple-harvesting robot. The harvesting robot is shown in Figure 8. The self-designed robot mainly includes three parts: a mobile carrier part, a 5-DOF (five-degree-of-freedom) manipulator part, and an end effector part.



(a) (b) (c)

(d) (e) (f)

Figure 6: The detection effect of Des-YOLO v4 algorithm. (a) Single fruit. (b) Multiple fruits. (c) Occluded fruit. (d) Overlapping fruits. (e) Backlight. (f) Sunlight.



(a)

(b)

Figure 7: Continued.

(c)

Figure 7: Detection comparison of different algorithms. (a) Faster R-CNN. (b) YOLO v4. (c) Des-YOLO v4.

Table 1: Performance comparison of different algorithms.

Algorithm	Backbone network	mAP (%)	Detection speed ($f{\cdot}s^{-1}$)
Faster R-CNN	ResNet50	88.1	9
YOLO v4	CSPDarknet53	87.9	53
Des-YOLO v4	Des-Darknet	93.1	51

Figure 8: Self-designed apple-harvesting robot.

The mobile carrier is crawler chassis, which is composed of chassis cabin and crawler walking mechanism. The chassis cabin is loaded with the environment sensing system and motion control unit of the harvesting robot. The crawler walking mechanism is composed of load-bearing wheel, driving wheel, tensioning auxiliary wheel, and belt supporting wheel. The 5-DOF manipulator adopts joint structure and is fixed on the mobile carrier. The first degree of freedom is the lifting platform, the second is the waist rotation joint of the manipulator, the third is the swing axis of the back arm, the fourth is the swing axis of the forearm and the fifth is the rotation axis of the robot end manipulator. The end effector adopts claw structure. The claw opening and closing is controlled by the stepper motor through the lead screw. The inner side of the clamping claw is equipped with pressure sensors, which can realize the lossless grasping of the apple.

In the harvesting experiment, the host computer of the robot first processes the apple images and detects the apple targets in the images through the Des-YOLO v4 algorithm. Then, the position of the target in the manipulator coordinate system is calculated. Finally, the manipulator is controlled to move toward the target by the visual servo control algorithm, so as to complete the apple-harvesting task. Figure 9 shows the complete process of robot harvesting operation. In this experiment, the fruit tree models are used to simulate the harvesting environment. A total of 70 harvesting experiments are carried out. The processing time of a single image is 0.4 seconds, the average single harvesting time is 8.7 seconds, and the comprehensive harvesting success rate is 92.9%. The Des-YOLO v4 algorithm can meet the real-time harvesting requirements of the harvesting robot.

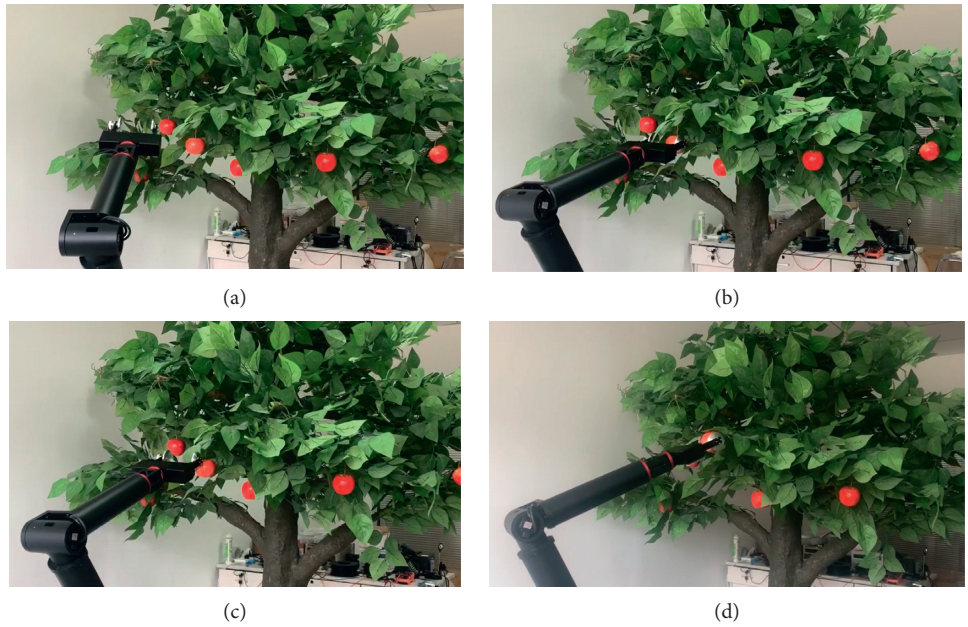


Figure 9: Harvesting experiment of apple-harvesting robot. (a) Starting position. (b) Looking for a target. (c) Detection process. (d) Successful capture.

4. Conclusions

This study proposed a Des-YOLO v4 algorithm and a detection method of apples. The algorithm can make the harvesting robots detect apples in complex environment. In addition, it has the advantages of higher recognition accuracy and faster detection speed compared with other detection algorithms.

The main conclusions are as follows:

(1) To improve the detection speed of harvesting robots, the Des-YOLO network structure is proposed. By adding the DenseNet, the parameters of YOLO v4 network are effectively reduced and the ability of the network to extract apple image features is improved. Therefore, the Des-YOLO network has better detection performance.

(2) Aiming at the problem of imbalance between positive and negative samples in the collected data, a class loss function based on AP-Loss is proposed. The AP-Loss function uses ranking task instead of classification task. It can improve the detection performance of the Des-YOLO v4 and improve the accuracy of apple recognition.

(3) In the test phase, Soft-NMS is used to replace NMS to solve the problem of missing apple detection, which improves the detection accuracy of apples under overlapping conditions.

(4) The Des-YOLO v4 algorithm is tested on the self-made apple data set. The test results show that the proposed algorithm has a mAP of 93.1% and a detection speed of 51 fps for apple images. Compared with Faster R-CNN and other network models, the proposed model can meet the accuracy and speed requirements of apple detection at the same time.

(5) A harvesting robot is designed to carry out the apple-harvesting experiment. The experimental results show that the processing time of a single image is 0.4 seconds, the single harvesting time is 8.7 seconds, and the comprehensive harvesting success rate is 92.9%.

However, the proposed algorithm still has some shortcomings. The network model in this study is still complex and needs a lot of computing time, which affects the overall picking efficiency. In low illumination environment, the performance of the algorithm will seriously descend, which makes the robot unable to work at night. Therefore, in the further research, the network model will be continued to reduce network parameters to improve the harvesting speed of the robot. Meanwhile, the detection method with night image will be studied, so that the harvesting robot can work in all illumination environments.

Data Availability

The Des-YOLO v4 model constructed in this study and datasets for training and evaluating the model are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

This research was supported by the Modern Agriculture Project of Jiangsu province (BE2020406) and the

International Science and Technology Cooperation Project of Zhenjiang City (GJ2020009).

References

[1] X. Wu, Z. Qi, and L. Wang, "Apple detection method based on lightweight YOLOv3 convolutional neural network," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 8, pp. 17–25, 2020.

[2] C. Zheng, P. Chen, J. Pang et al., "A mango picking vision algorithm on instance segmentation and key point detection from RGB images in an open orchard," *Biosystems Engineering*, vol. 206, no. 6, pp. 32–54, 2021.

[3] Y. Yu, K. Zhang, H. Liu, L. Yang, and D. Zhang, "Real-time visual localization of the picking points for a ridge-planting strawberry harvesting robot," *IEEE Access*, vol. 8, pp. 116556–116568, 2020.

[4] A. Kuznetsova, T. Maleva, and V. Soloviev, "Using YOLOv3 algorithm with pre- and post-processing for apple detection in fruit-harvesting robot," *Agronomy*, vol. 10, no. 7, pp. 10–16, 2020.

[5] Y. Tang, M. Chen, C. Wang et al., "Recognition and localization methods for vision-based fruit picking robots: a review," *Frontiers of Plant Science*, vol. 11, p. 510, 2020.

[6] D. Zhao, R. Wu, and X. Liu, "Design and experiment of a gas-electric hybrid drive all-weather apple harvesting robot," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 51, no. 02, pp. 28–35+43, 2020.

[7] J. Diniz, J. L. Ferreira, and P. Diniz, "A deep learning method with residual blocks for automatic spinal cord segmentation in planning CT," *Biomedical Signal Processing and Control*, vol. 71, no. 05, p. 103074, 2021.

[8] Y. Li, B. Fan, and W. Zhang, "Deep active learning for object detection," *Information Sciences*, vol. 579, no. 02, 2021.

[9] H. Wang, L. Dong, and H. Zhou, "YOLOv3-Litchi detection method of densely distributed litchi in large vision scenes," *Mathematical Problems in Engineering*, vol. 8883015, 2021.

[10] E. Kelman and R. Linker, "Vision-based localisation of mature apples in tree images using convexity," *Biosystems Engineering*, vol. 118, pp. 174–185, 2014.

[11] E. K. Nyarko, I. Vidović, and K. Cupec, "A nearest neighbor approach for fruit recognition in RGB-D images based on detection of convex surfaces," *Expert Systems with Applications*, vol. 114, no. DEC, pp. 454–466, 2018.

[12] W. Ji, X. Meng, and Y. Tao, "Fast segmentation of colour apple image under all-weather natural conditions for vision recognition of harvesting robots," *International Journal of Advanced Robotic Systems*, vol. 16, pp. 86-87, 2016.

[13] W. Jia, Y. Zheng, and D. Zhao, "Preprocessing method of night vision image application in apple-harvesting robot," *International Journal of Agricultural and Biological Engineering*, vol. 11, no. 2, pp. 54–56, 2018.

[14] Z. Song, Z. Zhou, W. Wang et al., "Canopy segmentation and wire reconstruction for kiwifruit robotic harvesting," *Computers and Electronics in Agriculture*, vol. 181, p. 105933, 2021.

[15] S. Khan, H. Rahmani, and S. A. Shah, "A Guide to convolutional neural networks for computer vision," *Synthesis Lectures on Computer Vision*, vol. 8, no. 1, pp. 02–13, 2018.

[16] A. Milella, R. Marani, A. Petitti, and G. Reina, "In-field high throughput grapevine phenotyping with a consumer-grade depth camera," *Computers and Electronics in Agriculture*, vol. 156, pp. 293–306, 2019.

[17] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, "DeepFruits: a fruit detection system using deep neural networks," *Sensors*, vol. 16, no. 8, p. 1222, 2016.

[18] M. Turan, Y. Almalioglu, and H. Araujo, "Deep EndoVO: a recurrent convolutional neural network (RCNN) based visual odometry approach for endoscopic capsule robots," *Neurocomputing*, vol. 275, pp. 737-738, 2018.

[19] Y. Ren, C. Zhu, and S. Xiao, "Object detection based on fast/faster RCNN employing fully convolutional architectures," *Mathematical Problems in Engineering*, vol. 2018, no. 9, pp. 1–7, 2018.

[20] X. Sun, P. Wu, and S. C. H. Hoi, "Face detection using deep learning: an improved faster RCNN approach," *Neurocomputing*, vol. 299, no. 19, pp. 42–50, 2018.

[21] J. Yang, W. He, and T. Zhang, "Research on subway pedestrian detection algorithms based on SSD model," *IET Intelligent Transport Systems*, vol. 14, no. 11, pp. 737-738, 2020.

[22] F. Liu, Y. Liu, and S. Lin, "Rapid identification method of tomato fruit in complex environment based on improved YOLO," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 51, no. 06, pp. 229–237, 2020.

[23] L. Fu, "Fast and accurate detection of kiwifruit in orchard using improved YOLOv3-tiny model," *Precision Agriculture*, 2020.

[24] Y. Xu, K. Imou, Y. Kaizu, and K. Saga, "Two-stage approach for detecting slightly overlapping strawberries using HOG descriptor," *Biosystems Engineering*, vol. 115, no. 2, pp. 144–153, 2013.

[25] C. Wang, T. Luo, and L. Zhao, "Window zooming–based localization algorithm of fruit and vegetable for harvesting robot," *IEEE Access*, vol. 7, no. 99, p. 1, 2019.

[26] L. Fu, Y. Feng, and Z. Liu, "Field multi cluster kiwifruit image recognition method based on convolution neural network," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 34, no. 02, pp. 205–211, 2018.

[27] J. Xiong, Z. Liu, and L. Tang, "Research on visual detection technology of green citrus in natural environment," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 49, no. 04, pp. 45–52, 2018.

[28] Y. Xue, N. Huang, and S. Tu, "Improved YOLO V2 recognition method for immature mango," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 34, no. 7, pp. 173–179, 2018.

[29] S. Inkyu, G. Zongyuan, and D. Feras, "Deep Fruits: a fruit detection system using deep neural networks," *Sensors*, vol. 16, no. 8, pp. 1222–1228, 2016.

[30] J. Xiong, Z. He, and L. Tang, "Visual positioning technology for disturbing grape picking point in unstructured environment," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 48, no. 04, pp. 29–33, 2017.

[31] Z. Chen, D. Ye, and C. Zhu, "Target recognition method based on improved YOLOv3," *Computer System Application*, vol. 29, no. 01, pp. 49–58, 2020.

[32] D. Wu, S. Lv, and M. Jiang, "Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments," *Computers and Electronics in Agriculture*, vol. 178, pp. 737-738, 2020.

[33] Q. Zhu, H. Zheng, Y. Wang, Y. Cao, and S. Guo, "Study on the evaluation method of sound phase cloud maps based on an improved YOLOv4 algorithm," *Sensors*, vol. 20, no. 15, p. 4314, 2020.

[34] T. T. Nguyen, K. Vandevoorde, N. Wouters, E. Kayacan, J. G. De Baerdemaeker, and W. Saeys, "Detection of red and bicoloured apples on tree with an RGB-D camera," *Biosystems Engineering*, vol. 146, pp. 33–44, 2016.

[35] Y. A. Liu, J. Xia, and B. Meng, "Extended dissipative synchronization for semi-Markov jump complex dynamic networks via memory sampled-data control scheme," *Journal of the Franklin Institute*, vol. 357, no. 15, 2020.

[36] K. Chen, W. Lin, J. Li et al., "AP-loss for accurate one-stage object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, p. 1, 2020.

[37] Y. Wang, X. Hu, K. Shi, X. Song, and H. Shen, "Network-based passive estimation for switched complex dynamical networks under persistent dwell-time with limited signals," *Journal of the Franklin Institute*, vol. 357, no. 15, pp. 10921–10936, 2020.