Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 8883015, 11 pages
https://doi.org/10.1155/2021/8883015

Research Article

YOLOv3-Litchi Detection Method of Densely Distributed Litchi in Large Vision Scenes

Hongjun Wang,[1] Lizhong Dong,[1,2] Hao Zhou,[1] Lufeng Luo,[3] Guichao Lin,[4] Jinlong Wu,[1] and Yunchao Tang[4]

[1]*Key Laboratory of Key Technology on Agricultural Machine and Equipment, Ministry of Education, South China Agricultural University, Guangzhou 510642, China*
[2]*Guangzhou Jiankun Network Technology Development Co. Ltd., Guangzhou 510530, China*
[3]*College of Mechanical and Electrical Engineering, Foshan University, Foshan 528000, China*
[4]*College of Urban and Rural Construction, Zhongkai University of Agriculture and Engineering, Guangzhou 510006, China*

Correspondence should be addressed to Lufeng Luo; luolufeng@fosu.edu.cn and Yunchao Tang; ryan.twain@gmail.com

Received 5 September 2020; Revised 20 December 2020; Accepted 21 January 2021; Published 4 February 2021

Academic Editor: Akhil Garg

Accurate and reliable fruit detection in the orchard environment is an important step for yield estimation and robotic harvesting. However, the existing detection methods often target large and relatively sparse fruits, but they cannot provide a good solution for small and densely distributed fruits. This paper proposes a YOLOv3-Litchi model based on YOLOv3 to detect densely distributed litchi fruits in large visual scenes. We adjusted the prediction scale and reduced the network layer to improve the detection ability of small and dense litchi fruits and ensure the detection speed. From flowering to 50 days after maturity, we collected a total of 266 images, including 16,000 fruits, and then used them to construct the litchi dataset. Then, the k-means++ algorithm is used to cluster the bounding boxes in the labeled data to determine the priori box size suitable for litchi detection. We trained an improved YOLOv3-Litchi model, tested its litchi detection performance, and compared YOLOv3-Litchi with YOLOv2, YOLOv3, and Faster R-CNN on the actual detection effect of litchi and used the F1 value and the average detection time as the assessed value. The test results show that the F1 of YOLOv3-Litchi is higher than that of YOLOv2 algorithm 0.1, higher than that of YOLOv3 algorithm 0.08, and higher than that of Faster R-CNN algorithm 0.05; the average detection time of YOLOv3-Litchi is 29.44 ms faster than that of YOLOv2 algorithm, 19.56 ms faster than that of YOLOv3 algorithm ms, and 607.06 ms faster than that of Faster R-CNN algorithm. And the detection speed of the improved model is faster. The proposed model remits optimal detection performance for small and dense fruits. The work presented here may provide a reference for further study on fruit-detection methods in natural environments.

1. Introduction

Litchi fruit has a high commercial value, but unfortunately, has a high tendency to drop from its mother plant to the ground as it grows. The setting percentage of litchi is significantly affected by environmental factors; the number of litchi fruits varies greatly in various environments. Smart agriculture necessitates accurately and efficiently collecting crop growth information. At present, there have been many studies on fruit detection. Robotic harvesting devices must detect fruits in the orchard and properly locate them to pick them [1, 2]; fruit detection can also automatically count the number of fruits in the field [3–7]. Automated fruit counting helps the orchard manager to measure fruit drop, estimate the yield, and plan for the market accordingly [8]. Machine-vision-based fruit detection technology is currently capable of detecting fruit growth information, providing early warnings for disease and pest infestations, yield prediction, harvest positioning, and other tasks. The use of robotics in orchards is increasing, particularly in yield prediction, yield mapping, and automated harvesting [9]. At the same time, machine vision, as the eyes of an intelligent robot, allows the

robot to perceive the operating environment, improve the robot's intelligence, and thereby improve its work efficiency and accuracy.

Early approaches to methodical fruit detection involved manually extracting the color, texture, contour, and other characteristics of the fruits. Lin et al. [10] detected fruits by a support vector machine classifier trained on color and texture features. Xu et al. [11] determined the area of strawberry fruits in images by calculating the color information of the HSV color space and then used the HOG feature combined with an SVM classifier to detect strawberries in the field. Lin et al. [12] proposed an algorithm for detecting spherical or cylindrical fruit of plants based on color, depth, and shape information, to guide the automatic picking of harvesting robots. Lu and Sang [13] proposed a citrus recognition method based on color information and contour segments; this involves a segmentation technique that combines color difference information and a normalized RGB model and then fits an ellipse to identify citrus under outdoor natural light. The preliminary segmentation stage does not exclude the influence of lighting in this case. Fu et al. [14] proposed an image processing method to segment and then separate linearly aggregated kiwi fruits. Li et al. [15, 16] used a reliable algorithm based on red-green-blue-depth (RGB-D) images to detect fruits.

There are also many important studies on litchi detection. He et al. [17] used AdaBoost to integrate multiple strong LDA classifiers to detect green litchi fruits in the natural environment. This method has good classification and recognition capabilities, but the classification takes too long to meet real-time requirements. Xiong et al. [18] proposed a method for identifying litchi at night and a calculation method for picking points. By using an improved fuzzy clustering method (FCM), the analysis method is combined with a one-dimensional random signal histogram to remove the background of the night scene image, and then, the Otsu algorithm is used to segment the fruit from the stem. This method requires high image quality and is more sensitive to noise. Wang et al. [19] used wavelet transform to normalize the image to reduce the influence of light and then used the K-means clustering algorithm to separate litchi fruits from branches and leaves. In the case of poor light conditions and serious fruit occlusion, this method has low recognition accuracy for mature tomato fruits. Guo et al. [20] presented a detection method based on monocular machine vision to detect litchi fruits growing in overlapped conditions. This method takes a long time to recognize and is not conducive to the picking efficiency of the robot. Fruit images taken in the natural environment often have variable lighting and complex backgrounds. Traditional algorithms can successfully identify litchi fruits, and its optimization method can reduce the impact of environmental changes on the detection results, but the robustness of traditional algorithms is limited.

Deep learning has gained popularity across various engineering fields in recent years [21–30]. Deep learning has been applied in the field of agriculture for pest identification, weed identification, and yield estimation [31]. Deep learning methods are generalized, robust, and suitable for detecting fruits in complex outdoor environments. There have been many studies on fruit detection based on deep learning in recent years [32–34]. Sa et al. [35] proposed a multimodal faster R-CNN which combines the RGB and NIR; compared with the previous bell pepper detection methods, the F1 score of sweet pepper detection increased from 0.807 to 0.838 and the speed was faster. Chen et al. [36] used deep learning to estimate the total number of fruits directly from an input picture; an FCN first segmented the fruit in the input image and then a counting neural network revealed an intermediate estimate of the number of fruits. Tian et al. [37] replaced the Darknet53 of YOLOv3 with DenseNet to improve feature propagation during training; the YOLOv3-dense model they trained detected apples at different growth stages more accurately and quickly than YOLO v3 or Faster RCNN. Koirala et al. [38] proposed the MangoYOLO network for real-time detection of mangoes in orchards with high accuracy and in real time. Wang et al. [39] used the Faster Region-based Convolutional Neural Network (R-CNN) model to identify fruits and vegetables. Gao et al. [40] proposed an apple detection method based on a fast regional convolutional neural network for multiclass apple dense fruit trees. Chen et al. [41] trained the robust semantic segmentation network for bananas and realized effective image preprocessing.

Deep learning is also used in the study of small-scale compacted fruits. Liang et al. [42] proposed a method to detect litchi fruits and fruiting body stems in a night environment. They detected litchi fruits in a natural environment at night based on YOLOv3 and then determined the region of interest of the fruit stems according to the bounding box of the litchi fruits (RoI) and finally segmented the fruit stems one by one based on U-Net, but the detection scheme in the complex orchard environment during the day has not yet been proposed. Santos et al. [43] used the latest convolutional neural network to successfully detect, segment, and track grape clusters to determine the shape, color, size, and compactness of grape clusters, but this method is not suitable for estimating the yield of grapes. In the actual field environment, the litchi fruits overlapped, blocked severely, and had different sparseness and different sizes. Therefore, the algorithm proposed in the above literature does not have a good solution for small and densely distributed litchi detection. These have become difficult points for rapid and accurate identification of litchi fruits.

The main contributions of this paper are as follows: (1) it proposes a densely distributed litchi detection algorithm YOLOv3-Litchi in a large visual scene. It adjusts the YOLOv3 prediction scale and reduces the network layer to improve the detection ability of small and dense litchi and ensure the detection speed. The final construction is a network model for tomato fruit recognition in a complex environment in the wild. (2) The proposed YOLOv3-Litchi algorithm was successfully trained and tested on the litchi data set and compared with the actual detection effect of YOLOv2, YOLOv3, and Faster R-CNN algorithm on litchi. The result shows the F1 value of the YOLOv3-Litchi algorithm, and the average detection time is better than the above algorithm, and it takes up less computer resources. (3)

YOLOv3-Litchi and YOLOv3 algorithms are used to detect litchi at different growth stages and compare the detection performance of the two at different growth stages. (4) In order to prove the robustness of the proposed algorithm, the YOLOv3-Litchi algorithm was used to detect litchi under strong light and weak light.

2. Materials and Methods

2.1. Data Preparation. As there is no public dataset for litchi fruit detection, we need to build our own dataset for training and testing for the task of litchi detection. In this paper, we collected images of litchi fruit trees at different stages of growth and constructed a litchi image dataset to support our fruit-recognition algorithm. The orchard where the images were collected is located on the South China Agricultural University campus, Guangzhou, Guangdong, China, at 113°21′ east longitude and 23°9′ north latitude. The image collection time was between 9 a.m. and 3 p.m. every Tuesday between April 17 and June 14. The distance between the camera (OLYMPUS E-M10 Mark III and 4/3 Live MOS image sensor) and that between the camera and the edge of the tree crown was 1-2 m during shooting, as shown in Figure 1. The resolution of those collected images is 4608×3072. In sunny weather, we adjusted the shooting angle to capture forward light and backlight. In cloudy weather, we took images under scattered light. The images collected included litchi fruit from 50 days after flowering to maturity.

In this paper, we have used 266 images to build the litchi image dataset. The fruits in these 266 images with a total of 16,000 targets were labeled in LabelImg software. Then, the 266 images were divided into a train set, valid set, and test set. The division results are shown in Table 1. All the litchi fruits in an image were labeled as one class.

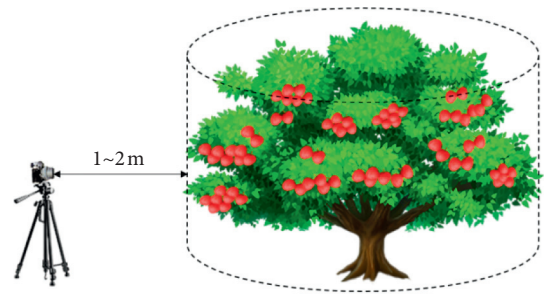

Figure 1: Image collection diagram.

Table 1: Dataset division details.

Dataset	Number of pictures	Number of fruits
Train set	210	11,390
Valid set	18	882
Test set	38	3,769

2.2. YOLOv3 Algorithm. YOLOv3 is evolved from YOLO and YOLOv2 [44–46]. YOLO series algorithms can directly predict the bounding box and corresponding class probability through a single neural network, making them faster than two-stage algorithms such as Faster RCNN [47]. The network structure of YOLOv3 is shown in Figure 2. YOLOv3 uses Darknet53 as the feature extraction network and uses the FPN structure to achieve the fusion of different scale features and multiple scale prediction. The use of multiple scale prediction makes YOLOv3 detect small targets better. Therefore, YOLOv3 is selected as the method to detect litchi fruit in this paper.

In the YOLOv3 algorithm, the original images are first resized to the input size, using a scale pyramid structure similar to the FPN network [48] and then divided into $S \times S$ grids according to the scale of the feature map. Take the input scale of 416×416 as an example, YOLOv3 will predict on three scales of feature map of 13×13, 26×26, 52×52, and use 2 times up-sampling to transfer features between 2 adjacent scales. In every prediction scale, every grid cell will predict 3 bounding boxes with the help of 3 anchor boxes. Therefore, the YOLOv3 network can be applied to the input pictures of any resolution size. As shown in Figure 3, if the center of a target falls into a grid, then the grid is responsible for predicting this target.

The network predicts 4 values for every bounding box on every grid, including the center coordinates (x, y) of the bounding box and the width w and height h of the target. YOLOv3 uses logistic regression to predict the confidence score of the target contained in the anchor box. The confidence score reflects whether the grid contains objects and the accuracy of the predicted location when the target is included. The confidence is defined as follows:

$$\text{confidence} = p_r(\text{object}) \times \text{IOU}_{\text{pred}}^{\text{truth}}, p_r(\text{object}) \in \{0, 1\}. \tag{1}$$

When the target is in the grid, $p_r(\text{object}) = 1$, otherwise 0. $\text{IOU}_{\text{pred}}^{\text{truth}}$ represents the consistency between the ground truth and the predicted box. If the confidence of the predicted bounding box is greater than a preset IoU threshold, the bounding box is retained. If multiple bounding boxes detect the same target, the best bounding box is selected by the NMS method.

2.3. Model Improvements. With the introduction of the FPN network, YOLOv3 takes good use of the high resolution of low-level features and the semantic information of high-level features, achieves the fusion of different levels via up-sampling and detects objects in three different prediction scales. The feature map with smaller prediction scale and larger receptive field is responsible for predicting bigger targets, while the feature map with a larger prediction scale and smaller receptive field is responsible for predicting smaller targets. For the fact that the targets in the litchi fruit dataset are generally small, adjusting the prediction scale of the module can improve the effect of detection.

Shallow information can be better utilized by adding a larger prediction scale. A larger feature map can then be obtained to enhance the detection ability for smaller fruits

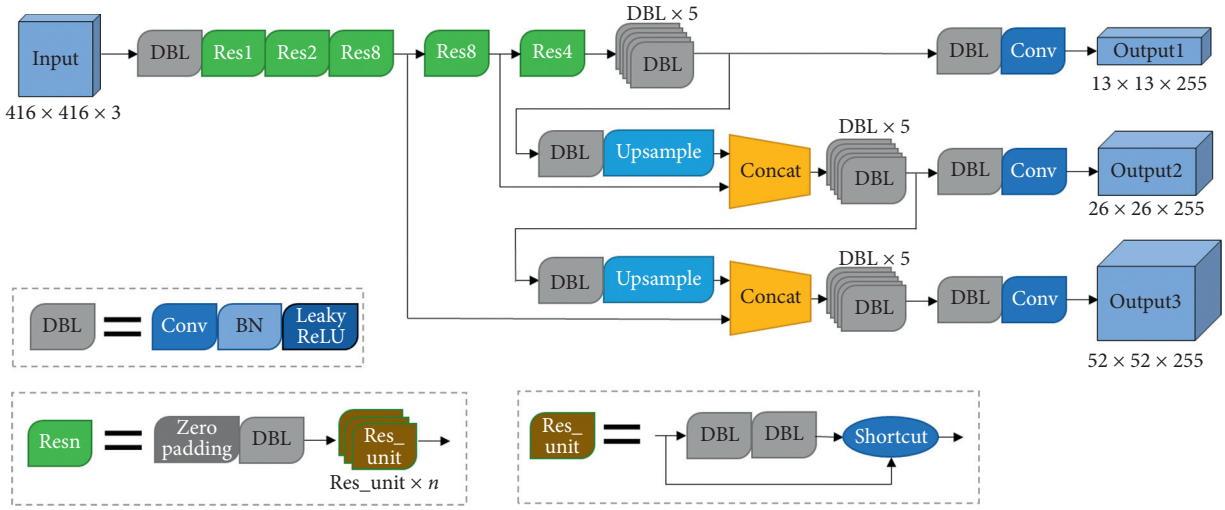

Figure 2: YOLOv3 network structure.

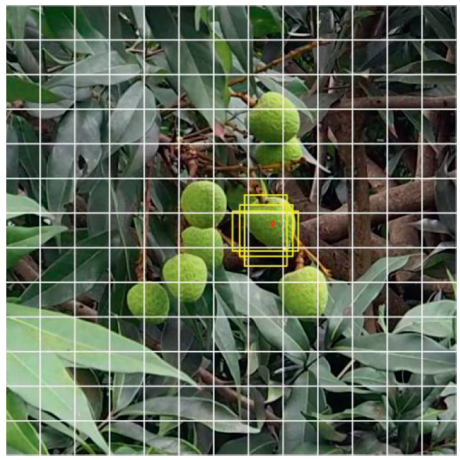

Figure 3: Bounding box prediction on 13 × 13 grids.

like litchis. The target litchi fruit is very small within the large scene, so smaller-scale feature outputs can be omitted.

In this study, we improved on the YOLOv3 model (Figure 4) to create a novel network structure. A new 104 × 104 scale feature map was output after upsampling the 52 × 52 feature map and merging it with a shallow 104 × 104 feature map. The 13 × 13 scale feature output and four residual units at the tail of the original Darknet53 were removed to form the proposed model, YOLOv3-Litchi.

2.4. Prior Box Clustering. Anchor mechanism is used in YOLOv3 to solve regression of the bounding box. YOLOv3 algorithm allocates 3 anchor boxes for every grid in every prediction scale, for 3 scales and a total of 9 anchor boxes. Although the network can adjust the size of the box, setting priori boxes helps the network learn features better. It is proposed that the k-means clustering method can be used to determine the size of prior boxes in YOLO. We clustered the labeled data to determine the size of the anchor boxes for litchi detection. The k-means clustering algorithm first randomly generates seed points and then performs cluster analysis based on the Euclidean distance. This clustering makes a larger box produce more errors than a smaller box. Our goal, in this case, is to obtain a larger IOU value between the anchor boxes and labeled boxes through clustering, so we used the following distance measure:

$$d_{(\text{box,centroid})} = 1 - \text{IOU}_{(\text{box,centroid})}. \quad (2)$$

The initial seed point of the k-means algorithm is randomly determined, which creates an unstable clustering result that is not globally optimal. We used the k-means ++ algorithm to solve this problem [49]. The k-means ++ algorithm is used to select initial seed points by maximizing the distance between the initial clustering centers.

The size of the anchor boxes clustered using the k-means algorithm and k-means ++ algorithm in this study is shown in Figure 5. The k-means ++ algorithm produced more diverse and stable results than the k-means algorithm. We ultimately determined nine anchor boxes in the model training process with the k-means ++ clustering algorithm.

3. Experiment and Discussion

The Darknet53 framework was used in this study to modify and train the proposed objection detection model. The models were trained on a computer running a 64-bit Ubuntu 16.04 system with Intel Core i7-7700K, 16 GB RAM, and NVIDIA GTX 1080Ti GPU.

Larger input sizes tend to produce better detection results in neural networks, but also means longer detection times. Multiscale training strategies can effectively improve the accuracy and robustness of the model. In this study, we trained a model with an input size of 416 × 416 through multiscale training [50]. We set the batch size to 64, the initial learning rate to 0.001, and the learning rate to 0.1 times the original after 15,000 steps and 20,000 steps. For the model with an input size of 416 × 416, we set nine anchor box sizes as follows: (3 × 5), (4 × 7), (6 × 9), (7 × 11), (9 × 13), (11 × 17), (14 × 22), (22 × 33), and (37 × 54). We

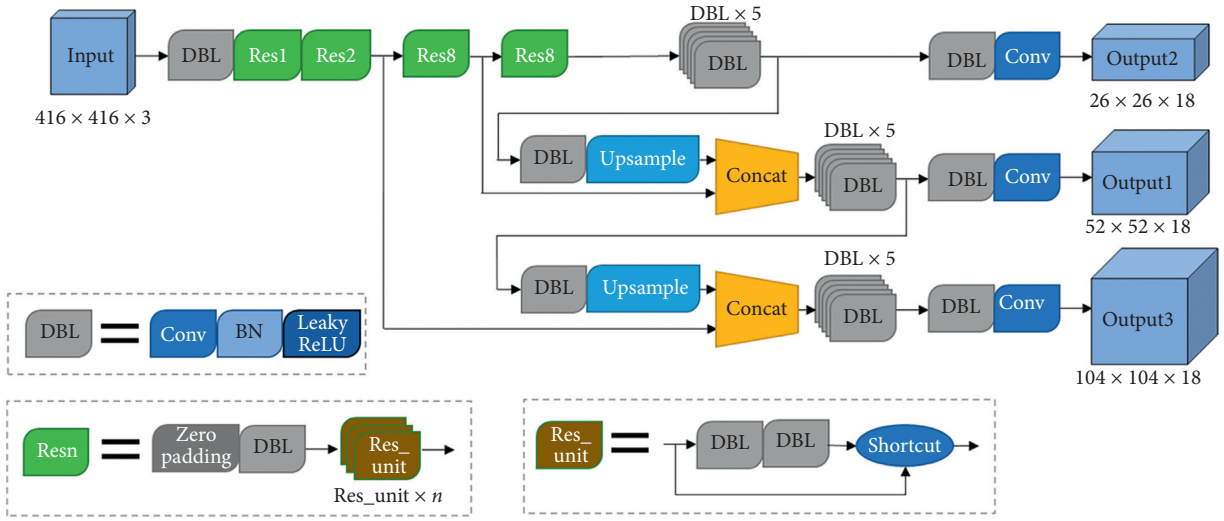

Figure 4: Proposed network structure.

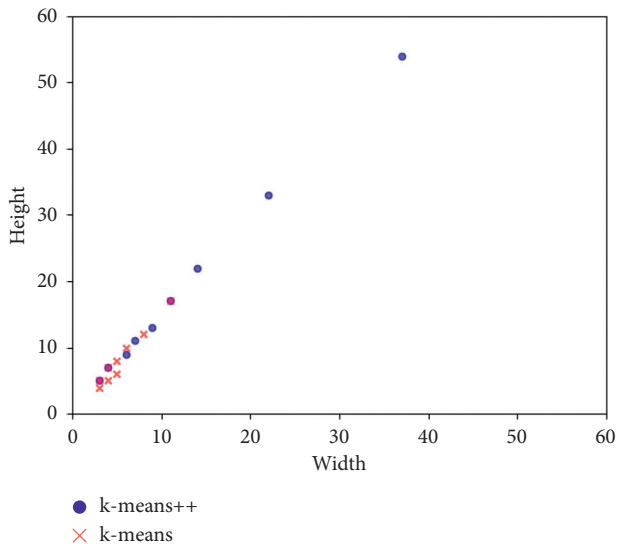

Figure 5: Results of two clustering methods.

trained the YOLOv3-Litchi model and the original YOLOv3 model to assess their performance by comparison between them.

3.1. Related Evaluation Indicators. Intersection-over-union (IoU) is an important parameter when verifying object detection results. It represents the ratio of the intersection and union between the "predicted bound" and "ground truth bound," as shown in equation (3). If the IoU of the detection result exceeds a given threshold, the result is correct; otherwise, it is incorrect.

$$\text{IoU} = \frac{\text{area}(P) \cap \text{area}(G)}{\text{area}(P) \cup \text{area}(G)}. \tag{3}$$

Common evaluation indicators for object detection include precision, recall, F1 score, and mean average precision (mAP). The results were predicted by the model include true-positive samples (TP), false-positive samples (FP), and false-negative samples (FN). Precision is defined as

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{4}$$

The recall is defined as

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{5}$$

F1 score is defined as

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{6}$$

The precision is the proportion of all samples that are predicted to be a certain category. The mAP is an indicator that reflects global performance. AP is the average precision rate, and mAP is the average value of the APs of all classes. The recall is the proportion of all samples whose true label is a certain category. F1 score is a comprehensive index, which is the harmonic average of precision and recall.

We plotted the models' precision-recall (P-R) curves, where recall is the horizontal axis and precision is the vertical axis, to compare their performance of the models. The P-R curve intuitively reflects the precision and recall of a model on an overall sample. If a curve protrudes more to the upper right, then its corresponding model is more effective.

3.2. Comparison of Results. To validate the performance of the proposed YOLOv3-Litchi model, other state-of-the-art detection methods were evaluated for comparison—YOLOv2, YOLOv3, and Faster R-CNN. Faster R-CNN is a detection method based on a region generation network. This method first generates a series of sample candidate frames by an algorithm and then classifies samples through a convolutional neural network. YOLOv2 and YOLOv3 are regression-based detection methods. The method does not need to generate candidate frames, directly converts the problem of target frame positioning into regression problem

processing, and predicts target classification while achieving target positioning. The test set contains a total of 38 images each 4608 × 3072 pixels in size. We set the detection confidence threshold to 0.25 and the IoU threshold to 0.5. The model receives images of 416 × 416 pixels as inputs. The results are given in Table 2.

As shown in Table 2, under the same output size, the detection performance of the proposed model is significantly better than other models. Compared with YOLOv2, YOLOv3-Litchi's model accuracy has increased by 0.07, the recall rate has increased by 0.11, the F1 score has increased by 0.1, and the mAP has increased by 0.17; compared with YOLOv3, the model's accuracy has increased by 0.06, the recall rate has increased by 0.09, the F1 score increased by 0.08, and the mAP increased by 0.15; compared with Faster R-CNN, the accuracy of the model increased by 0.03, the recall rate increased by 0.06, the F1 score increased by 0.05, and the mAP increased by 0.12.

The P-R curves of each model are shown in Figure 6. When the input dimensions are the same, the P-R curve of the proposed model is more convex to the upper right, indicating better performance.

Figure 7 shows the detection results of the YOLOv3-Litchi model. Because the target is small, it is difficult to observe. To effectively compare the detection results of the two models we tested, we isolated a portion of the detection result image as shown in Figure 8.

As shown in Figure 8, we compared the detection results in the case of small, densely distributed fruits–some of which were not detected by any model. Under the same input size, the proposed model did appear to detect litchis most accurately. When the input size of the model is larger, even more litchis could be detected.

3.2.1. Comparison of Detection Time. The average detection time of the four cases was also tested for comparison against the two models (Figure 9).

The test results show that the average detection time of YOLOv3-Litchi is 29.44 ms faster than that of YOLOv2 algorithm, 19.56 ms faster than that of YOLOv3 algorithm, and 607.06 ms faster than that of Faster R-CNN algorithm. The proposed YOLOv3-Litchi model has the fastest detection speed, which indicates that the model can perform litchi detection in real time, which is important for harvesting robots.

3.2.2. Detection of Litchis at Different Growth Stages. The size and density of litchis differ in different stages of growth. We collected 12 images of young litchis, 12 images of expanding litchis, and 9 images of ripe litchis from the test set to compare the detection results of the proposed and original model among them. Table 3 shows the detection effects of the models with input size of 416 × 416.

The detection results for young litchis, expanding litchis, and ripe litchis between the proposed and original models are shown in Figures 10–12. The upper left corner of each figure is an enlarged view of the clustered litchi parts.

Table 2: Test results of 4 models.

Model name	Precision (%)	Recall (%)	F1 score	mAP (%)
YOLOv2	80.54	63.92	0.71	60.73
YOLOv3	81.72	65.96	0.73	62.34
Faster R-CNN	84.10	68.75	0.76	65.24
YOLOv3-Litchi	87.43	74.93	0.81	77.46

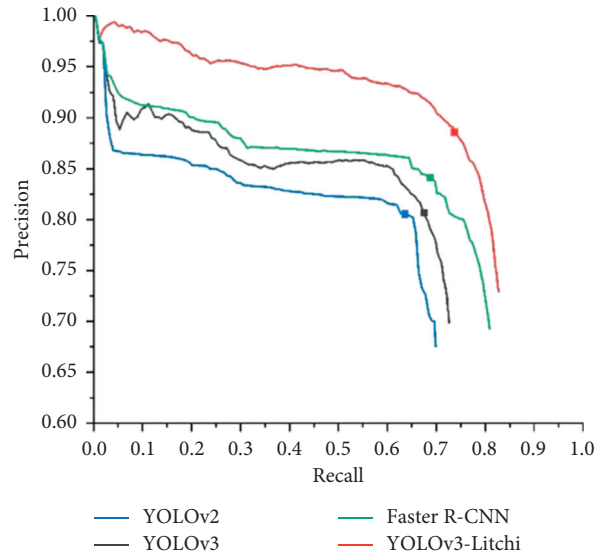

Figure 6: P-R curve.

The results show that YOLOv3-Litchi can also successfully detect litchi at different growth stages, and the detection performance at each growth stage is better than YOLOv3; the proposed algorithm shows the worst detection performance on young litchi, followed by expanding litchi. It performs best on ripe litchi.

3.2.3. Detection under Different Illumination Conditions. In the natural environment, dynamic weather changes and the sun's movement across the field create continuous changes in the illumination on the litchi tree. The brightness of the fruit image changes with these changes in illumination, which affects the fruit-detection algorithm.

Figure 13 shows where the YOLOv3-Litchi model detects litchis under both strong and weak light illumination conditions.

3.2.4. Model Size. The size of the model files is shown in Table 4. As the proposed model has fewer network layers and fewer parameters, it consumes less memory than the original model. The file size of the improved model is 76.9 MB, which is about 1/3 that of the original YOLOv3 model. However, Faster R-CNN is a two-stage target detection algorithm. It first generates a series of candidate frames as samples by the algorithm and then classifies the samples through the convolutional neural network, which occupies more memory than other algorithms.

Figure 7: Prediction results on one image.

(a) (b) (c) (d)

Figure 8: Comparison of local detection results. (a) YOLOv2 and (b) YOLOv3 results; (c) Faster R-CNN and (d) YOLOv3-Litchi results.

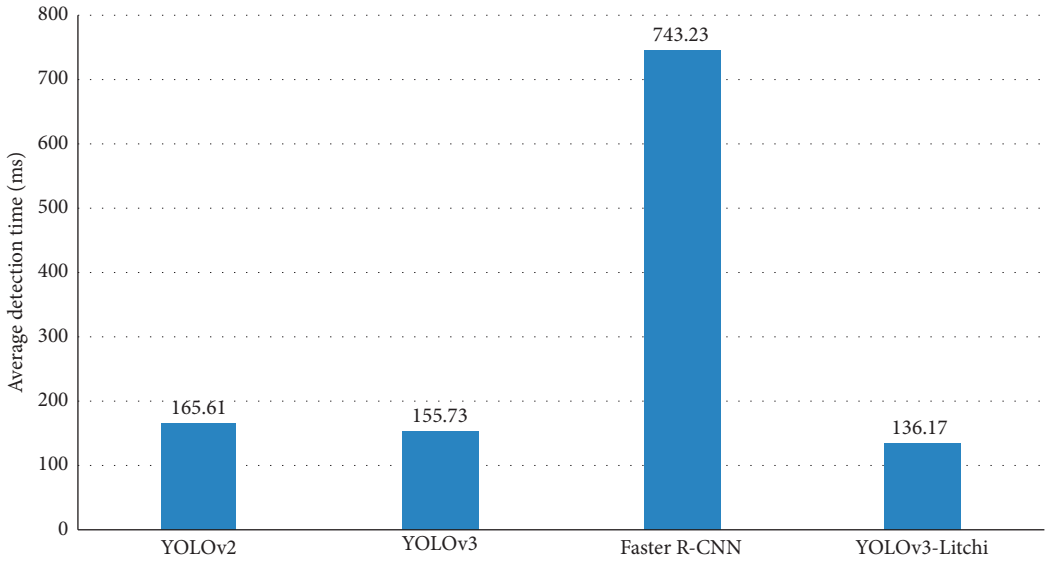

Figure 9: Fruit detection speed.

Table 3: Test results of litchis at different growth stages.

Model	Growth stage	Precision (%)	Recall (%)	F1 score	mAP (%)
YOLOv3	Young litchi	75.31	53.43	0.63	47.87
	Expanding litchi	79.95	56.48	0.66	55.48
	Ripe litchi	86.58	81.45	0.84	83.49
YOLOv3-Litchi	Young litchi	84.15	64.38	0.73	67.26
	Expanding litchi	86.44	67.33	0.76	71.86
	Ripe litchi	91.40	85.98	0.89	93.76

(a) (b)

Figure 10: Detection results on young litchis: (a) original YOLOv3 model and (b) proposed YOLOv3-Litchi.

(a) (b)

Figure 11: Detection results of expanding litchis: (a) original YOLOv3 model and (b) proposed YOLOv3-Litchi.

(a) (b)

Figure 12: Detection results on ripe litchis: (a) original YOLOv3 model and (b) proposed YOLOv3-Litchi.

(a) (b)

Figure 13: Detection results under different illumination: (a) litchis under strong light and (b) litchis under weak light.

Table 4: Comparison of model sizes.

Model	Size (MB)
YOLOv2.weights	268
YOLOv3.weights	234
YOLOv3-Litchi.weights	76.9

4. Conclusions

An improved YOLOv3 model was established in this study for automatic detection of small-scale and densely growing litchi fruits in large orchard scenes. Our conclusions can be summarized as follows:

(1) We adjusted the output scale of the original YOLOv3 network and reduced its depth to build the proposed improved YOLOv3-Litchi model. We used the k-means++ clustering algorithm to cluster the bounding boxes, obtaining nine prior boxes for litchi detection model training.

(2) The proposed model and the original model were trained and tested on the litchi dataset. The results show that YOLOv3-Litchi can successfully detect litchi, which is suitable for small and densely distributed fruits.

(3) When the input size is 416×416, the F1 score of the YOLOv3-Litchi model for litchi detection is 0.81 and the average detection time for a single image is 136.17 ms. The proposed model can effectively detect litchi under strong and weak lighting conditions. Compared with other network models, this model takes into account the requirements of recognition accuracy and speed and has the highest detection and positioning accuracy and the best comprehensive performance.

(4) YOLOv3-Litchi and YOLOv3 algorithms have been used to successfully detect litchi in different growth stages. YOLOv3-Litchi has better detection performance at each growth stage than YOLOv3; the proposed algorithm shows the worst detection performance on young litchi, followed by expanding litchi perform best on ripe litchi.

(5) We used the YOLOv3-Litchi algorithm to compare the actual detection effect of litchi with YOLOv2, YOLOv3, and Faster R-CNN and used the F1 value and the average detection time as the evaluation value. The test results show that F1 of YOLOv3-Litchi is higher than that of YOLOv2 algorithm 0.1, higher than that of YOLOv3 algorithm 0.08, and higher than that of Faster R-CNN algorithm 0.05; the average detection time of YOLOv3-Litchi is 29.44 ms faster than that of YOLOv2 algorithm, 19.56 ms faster than that of YOLOv3 algorithm, and 607.06 ms

faster than that of Faster R-CNN algorithm. YOLOv3-Litchi occupies the least computer resources than other algorithms.

The method proposed in this paper may serve as a workable reference for further research on dense fruit detection in large visual scenes.

Although the YOLOv3-Litchi model proposed in this paper can detect litchi fruit with dense distribution well, the current detection is tested with sharp pictures and the sample size is limited. In our future work, we will collect more data, build a larger dataset for training, and study the dynamic detection of litchi fruit in monitoring images in the natural environment.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by grants from the National Natural Science Foundation of China (no. 51705365), the Special Fund for Rural Revitalization Strategy of Guangdong Province (no. 2018A0169), the Key-Area Research and Development Program of Guangdong Province (no. 2019B020223003), the Science and Technology Planning Project of Guangdong Province (no. 2019A050510035), and the Research Projects of Universities Guangdong Province (no. 2019KTSCX197).

References

[1] Y. Tang, M. Chen, C. Wang, L. Luo, J. Li, and X. Zou, "Recognition and localization methods for vision-based fruit picking robots: a review," *Front Plant Sci*, vol. 11, p. 510, 2020.

[2] Y. Zhao, L. Gong, Y. Huang, and C. Liu, "Robust tomato recognition for robotic harvesting using feature images fusion," *Sensors*, vol. 16, no. 2, p. 173, 2016.

[3] W. S. Qureshi, A. Payne, K. B. Walsh, R. Linker, O. Cohen, and M. N. Dailey, "Machine vision for counting fruit on mango tree canopies," *Precision Agriculture*, vol. 18, no. 2, pp. 224–244, 2017.

[4] Y. Song, C. A. Glasbey, G. W. Horgan, G. Polder, J. A. Dieleman, and G. W. A. M. van der Heijden, "Automatic fruit recognition and counting from multiple images," *Biosystems Engineering*, vol. 118, pp. 203–215, 2014.

[5] S. Bargoti and J. P. Underwood, "Image segmentation for fruit detection and yield estimation in apple orchards," *Journal of Field Robotics*, vol. 34, no. 6, pp. 1039–1060, 2017.

[6] C. Wang, W. S. Lee, X. Zou, D. Choi, H. Gan, and J. Diamond, "Detection and counting of immature green citrus fruit based on the Local Binary Patterns (LBP) feature using illumination-normalized images," *Precision Agriculture*, vol. 19, no. 6, pp. 1062–1083, 2018.

[7] Y. Chen, W. S. Lee, H. Gan et al., "Strawberry yield prediction based on a deep neural network using high-resolution aerial orthoimages," *Remote Sensing*, vol. 11, no. 13, p. 1584, 2019.

[8] Z. Wang, K. Walsh, and A. Koirala, "Mango fruit load estimation using a video based MangoYOLO-kalman filter-Hungarian algorithm method," *Sensors*, vol. 19, no. 12, p. 2742, 2019.

[9] L. Fu, F. Gao, J. Wu, R. Li, M. Karkee, and Q. Zhang, "Application of consumer RGB-D cameras for fruit detection and localization in field: a critical review," *Computers and Electronics in Agriculture*, vol. 177, p. 105687, 2020.

[10] G. Lin, Y. Tang, X. Zou, J. Cheng, and J. Xiong, "Fruit detection in natural environment using partial shape matching and probabilistic Hough transform," *Precision Agriculture*, vol. 21, no. 1, pp. 160–177, 2020.

[11] 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.

[12] G. Lin, Y. Tang, X. Zou, J. Xiong, and Y. Fang, "Color-, depth-, and shape-based 3D fruit detection," *Precision Agriculture*, vol. 21, no. 1, pp. 1–17, 2020.

[13] J. Lu and N. Sang, "Detecting citrus fruits and occlusion recovery under natural illumination conditions," *Computers and Electronics in Agriculture*, vol. 110, pp. 121–130, 2015.

[14] L. Fu, E. Tola, A. Al-Mallahi, R. Li, and Y. Cui, "A novel image processing algorithm to separate linearly clustered kiwifruits," *Biosystems Engineering*, vol. 183, pp. 184–195, 2019.

[15] G. Lin, Y. Tang, X. Zou, J. Li, and J. Xiong, "In-field citrus detection and localisation based on RGB-D image analysis," *Biosystems Engineering*, vol. 186, pp. 34–44, 2019.

[16] J. Li, Y. Tang, X. Zou, G. Lin, and H. Wang, "Detection of fruit-bearing branches and localization of litchi clusters for vision-based harvesting robots," *IEEE Access*, vol. 8, pp. 117746–117758, 2020.

[17] Z.-L. He, J.-T. Xiong, R. Lin et al., "A method of green litchi recognition in natural environment based on improved LDA classifier," *Computers and Electronics in Agriculture*, vol. 140, pp. 159–167, 2017.

[18] J. Xiong, R. Lin, Z. Liu et al., "The recognition of litchi clusters and the calculation of picking point in a nocturnal natural environment," *Biosystems Engineering*, vol. 166, pp. 44–57, 2018.

[19] C. Wang, Y. Tang, X. Zou, L. Luo, and X. Chen, "Recognition and matching of clustered mature litchi fruits using binocular charge-coupled device (CCD) color cameras," *Sensors*, vol. 17, no. 11, p. 2564, 2017.

[20] Q. Guo, Y. Chen, Y. Tang et al., "Lychee fruit detection based on monocular machine vision in orchard environment," *Sensors*, vol. 19, no. 19, p. 4091, 2019.

[21] T. Hussain, S. M. Siniscalchi, C.-C. Lee, S.-S. Wang, Y. Tsao, and W.-H. Liao, "Experimental study on extreme learning machine applications for speech enhancement," *IEEE Access*, vol. 5, pp. 25542–25554, 2017.

[22] C. Songnan, T. Mengxia, and K. Jiangming, "Predicting depth from single RGB images with pyramidal three-streamed networks," *Sensors (Basel, Switzerland)*, vol. 19, no. 3, p. 667, 2019.

[23] B. Jiang, H. Song, and D. He, "Lameness detection of dairy cows based on a double normal background statistical model," *Computers and Electronics in Agriculture*, vol. 158, pp. 140–149, 2019.

[24] Y. Wu, X. Hu, Z. Wang, J. Wen, J. Kan, and W. Li, "Exploration of feature extraction methods and dimension for

sEMG signal classification," *Applied Sciences*, vol. 9, no. 24, p. 5343, 2019.

[25] S. Oh, S. Oh, and B. Min, "Development of high-efficiency, high-speed and high-pressure ambient temperature filling system using pulse volume measurement," *Applied Sciences*, vol. 9, no. 12, p. 2491, 2019.

[26] Q. Zhang and G. Gao, "Prioritizing robotic grasping of stacked fruit clusters based on stalk location in RGB-D images," *Computers and Electronics in Agriculture*, vol. 172, p. 105359, 2020.

[27] B. Jiang, Q. Wu, X. Yin, D. Wu, H. Song, and D. He, "FLYOLOv3 deep learning for key parts of dairy cow body detection," *Computers and Electronics in Agriculture*, vol. 166, p. 104982, 2019.

[28] L. Fu, Y. Feng, J. Wu et al., "Fast and accurate detection of kiwifruit in orchard using improved YOLOv3-tiny model," *Precision Agriculture*, 2020.

[29] L. Fu, Y. Majeed, X. Zhang, M. Karkee, and Q. Zhang, "Faster R-CNN-based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting," *Biosystems Engineering*, vol. 197, pp. 245–256, 2020.

[30] G. Lin, Y. Tang, X. Zou, J. Xiong, and J. Li, "Guava detection and pose estimation using a low-cost RGB-D sensor in the field," *Sensors*, vol. 19, no. 2, p. 428, 2019.

[31] A. Koirala, K. B. Walsh, Z. Wang, and C. McCarthy, "Deep learning - method overview and review of use for fruit detection and yield estimation," *Computers and Electronics in Agriculture*, vol. 162, pp. 219–234, 2019.

[32] Q. Liang, W. Zhu, J. Long, Y. Wang, W. Sun, and W. Wu, "A real-time detection framework for on-tree mango based on SSD network," in *Proceedings of the 11th International Conference, ICIRA 2018*, Newcastle, Australia, August 2018.

[33] H. Basri, I. Syarif, and S. Sukaridhoto, "Faster R-CNN implementation method for multi-fruit detection using tensorflow platform," in *Proceedings of the 2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing*, pp. 337–340, Surabaya, Indonesia, October 2018.

[34] N. Lamb and M. C. Chuah, "A strawberry detection system using convolutional neural networks," in *Proceedings of the IEEE International Conference on Big Data*, pp. 2515–2520, Seattle, WA, USA, December 2018.

[35] 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.

[36] S. W. Chen, S. S. Shivakumar, S. Dcunha et al., "Counting apples and oranges with deep learning: a data-driven approach," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 781–788, 2017.

[37] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, "Apple detection during different growth stages in orchards using the improved YOLO-V3 model," *Computers and Electronics in Agriculture*, vol. 157, pp. 417–426, 2019.

[38] A. Koirala, K. B. Walsh, Z. Wang, and C. McCarthy, "Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of "MangoYOLO"" *Precision Agriculture*, vol. 20, no. 6, p. 1107, 2019.

[39] C. Wang, T. Luo, L. Zhao, Y. Tang, and X. Zou, "Window zooming-based localization algorithm of fruit and vegetable for harvesting robot," *IEEE Access*, vol. 7, pp. 103639–103649, 2019.

[40] F. Gao, L. Fu, X. Zhang et al., "Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN," *Computers and Electronics in Agriculture*, vol. 176105634 pages, 2020.

[41] M. Chen, Y. Tang, X. Zou et al., "Three-dimensional perception of orchard banana central stock enhanced by adaptive multi-vision technology," *Computers and Electronics in Agriculture*, vol. 174, p. 105508, 2020.

[42] C. Liang, J. Xiong, Z. Zheng et al., "A visual detection method for nighttime litchi fruits and fruiting stems," *Computers and Electronics in Agriculture*, vol. 169, p. 105192, 2020.

[43] T. T. Santos, L. L. de Souza, A. A. dos Santos, and S. Avila, "Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association," *Computers and Electronics in Agriculture*, vol. 170, p. 105247, 2020.

[44] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[45] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, IEEE, New York, NY, USA, 2016.

[46] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2016.

[47] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, 2017.

[48] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2016.

[49] D. Arthur and S. Vassilvitskii, *K-Means: The Advantages of Careful Seeding*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.