Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 4648105, 10 pages
https://doi.org/10.1155/2022/4648105

Research Article

A Dense Litchi Target Recognition Algorithm for Large Scenes

Jinlong Wu,[1] Sheng Zhang,[2] Tianlong Zou,[3] Lizhong Dong,[1] Zhou Peng, and Hongjun Wang[1]

[1]*South China Agricultural University, College of Engineering, Guangzhou 510642, China*
[2]*Guangdong RuoBo Intelligent Co., Ltd., Foshan 528200, China*
[3]*Foshan-Zhongke Innovation Research Institute of Intelligent Agriculture and Robotics, College of Engineering, Foshan 528200, China*

Correspondence should be addressed to Hongjun Wang; xtwhj@scau.edu.cn

Received 11 August 2021; Revised 28 December 2021; Accepted 14 March 2022; Published 7 April 2022

Academic Editor: Mohammad Yaghoub Abdollahzadeh Jamalabadi

To address the automatic detection of dense and small-scale fruit targets under natural large-scene conditions, litchi was used as the research object. Here, a method to automatically detect dense and small-scale litchi fruit targets based on the YOLOv4 detection network is proposed. First, the K-means++ algorithm was used to cluster the labelled data frames (ground truth) to determine the size of the anchor suitable for litchi. Then, the output size of the feature map of the original network was changed to make it more suitable for small-scale target detection. In addition, the images were preprocessed (cropped input) before they were fed into the network. To construct the litchi dataset, 400 images containing more than 20,000 targets were collected. Comparing the detection level to that of the original YOLOv4 model, the recall, precision, and F1 score values of the improved model increased from 0.81 to 0.825, 0.762 to 0.892, and 0.79 to 0.85, respectively. The experimental results indicate that the performance of the litchi detection method proposed in the study is significantly greater than the original model, and it meets the requirements for fruit monitoring in litchi orchards.

1. Introduction

Litchi is a characteristic fruit of south of the Five Ridges. It not only produces a large amount of fruit but also has a high fruit-drop rate [1, 2]. Monitoring the litchi growth process and scientifically managing the planting process helps fruit farmers predict yields and correctly develop market plans (e.g., pricing products and hiring manpower). In recent years, the detection of litchi fruit in the natural environment using machine vision methods has received widespread attention. With the development of Internet of Things (IoT) technology, the concepts of unmanned farms and precision agriculture have been proposed [3]. Unlike traditional agriculture, unmanned farms allow farmers to access information about their orchards without leaving their homes and to make appropriate decisions based on the information obtained [4–9]. This enables truly intelligent scientific management. The method of real-time monitoring of orchards through fruit recognition by in-orchard cameras has gained widespread attention. Accurate fruit identification has many uses. It provides useful information for fruit harvesting, ripeness detection [10–12], and fruit yield prediction. Fruit farmers are able to develop scientific management and marketing plans through yield estimation [13–17]. Additionally, with the machine harvesting of fruit, fruit recognition provides information for the accurate positioning of the robot [18–25].

Initially, fruit recognition used mainly traditional image recognition algorithms. For example, Zhou et al. used texture-based edge detection combined with red measurement, region thresholding, and circle fitting to identify apples in images. Lu et al. used grayscale images with chromatic aberration GB and the Otsu algorithm to segment the citrus and background, and then, they fitted the contours by least squares circle fitting with Tukey's weight function [26]. Liu et al. first applied matching expansion in peach-growing areas to identify the whole region, and then, they fitted the centroid and radius of the peach by calculating the

statistical parameters of the potential centroids [27]. Fu et al. analyze the images in HSV space to remove parts of the background. A support vector machine having local binary pattern features and a histogram of banana-oriented gradient features were then used to find the banana region [28]. Yu et al. used color and texture features to train a random forest binary classification model to identify litchi fruit [29].

With continuous development and progress, many classical algorithms for target detection based on deep learning have been proposed. Unlike traditional image recognition methods, deep learning can discover and carve out the complex structural features inside the issue, instead of manually extracting features, which can greatly improve the algorithm's detection performance. Many deep learning algorithms for fruit detection have been proposed. They can be mainly divided into two categories. One class is the two-stage detector represented by Fast R-CNN [30] and Faster R-CNN [31]. The other class is the single-stage detector represented by SSD [32] and You Only Look Once (YOLO) [33–35]. This type of model does not have a separate regional extraction network and relies mainly on a predefined prior frame (anchor) for regression on the target. For the difficult problem of automatically detecting litchi targets under natural large-scene conditions, we propose a method based on the YOLOv4 deep learning algorithm to detect small-scale dense distributions of litchi fruit.

The main contribution of our method is the improved YOLOv4-based detection model, which makes the model more suitable for detecting small and dense targets such as litchi. A cropping chunking recognition method is also used to further improve the accuracy of detecting small targets. These methods are targeted at solving the difficult problem of automatically detecting litchi targets under natural large-scene conditions with good results. It effectively improves the detection accuracy of litchi by the YOLOv4 model under natural large-scene conditions.

Initially, we modified the backbone of YOLOv4. The extraction of a 32-fold downsampled feature map was eliminated, and the output detection of the original large-scale feature map was removed. Then, the K-means++ algorithm was used to obtain the anchor box by clustering the litchi dataset. Finally, using the chunk cropping method, the images were cropped and inputted into the network separately for detection before they are all inputted into the network. After all the detection results are filtered using a nonmaximum suppression (NMS) algorithm, they are mapped back to the original image. Additionally, we used common detection metrics (Precision, Recall, and F1) to determine the performance of the model. The model results were compared with those of the original YOLOv4 model and the image of the actual scene.

2. Materials and Methods

2.1. Dataset Preparation. The dataset used in this study contains images of litchi trees collected in an orchard within the South China Agricultural University. The time periods for collecting the images included morning and afternoon. The resolution of the collected images was $4{,}608 \times 3{,}072$. Images of both ripe red and immature green litchi were collected.

A total of 400 images were collected, containing more than 20,000 targets. The dataset was divided into training and test sets. LabelImg software was used to label the litchi fruit and generate an XML file containing the target type and location. It was then converted into a data format suitable for the YOLOv4 network. The data division results are shown in Table 1.

2.2. Yolov4 Algorithm. YOLO is a deep learning target detection algorithm based on regression. Its main idea is to convert the problem of object detection into a solvable regression problem. The YOLOv4 model detection process is as follows. First, the input image is resized to the input size. Then, the downsampling process is performed five times and the last three downsampling results are saved to predict the target. Then, the input image is divided into feature maps of $S \times S$ cells, and the subsequent output is predicted in units of cells.

YOLOv4 establishes a total of nine groups of anchors and three outputs, and each output is assigned three groups of anchors. Consequently, each cell of the output predicts three bounding boxes (Figure 1).

2.3. Improved Model Lit-YOLO. Deeper convolutional neural networks have a stronger nonlinear representation and can learn to fit more complex features. However, as the network becomes deeper, problems, such as increased network computations, slower inferences, and feature disappearance, may occur. Moreover, the detection used here, litchi, is a small target that lacks rich semantic information. The original YOLOv4 network outputs three scales of feature maps for target prediction. The feature map with the smallest resolution (26×26) has severe semantic losses and has the worst performance for detecting small targets. Consequently, the Lit-YOLO model based on YOLOv4 was proposed. The extraction of the 32-fold downsampling in the original backbone network (Bone) was removed. The output detection of the original network on the original large-scale feature map was removed. To allow the network to acquire more feature information from small targets and improve the detection rate of small targets, we used the fourfold downsampled feature map output in the original network for target detection because it contains more information about the locations of small targets. The semantic information from the 16-fold downsampling was merged with the shallow network by upsampling to convey powerful semantic features from top to bottom. The model further improved the feature extraction capability, enhanced the detection accuracy of small targets, and reduced the amount of model computation. The improved network structure is shown in Figure 2.

2.4. Clustering Algorithm. Many target detection algorithms (e.g., YOLO, RCNN, and SSD) solve the edge regression problem through the anchor box mechanism. The so-called

Table 1: Dataset division information.

Dataset	Number of images	Percentage (%)
Train set	320	80
Test set	80	20

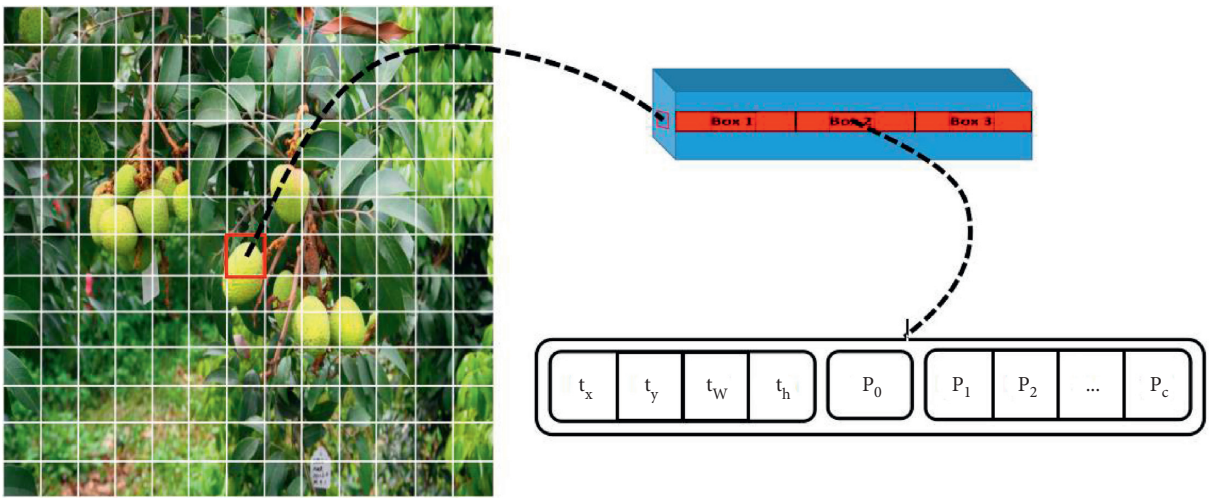

Figure 1: YOLOv4 network output. The parameters contained in each bounding box are as follows: the center coordinates of the box (t_x, t_y), the width (t_w), and height (t_h) of the target, the confidence score (P_0) that the box has a target, and the probability (P_c) that the target in the box belongs to each type of object.

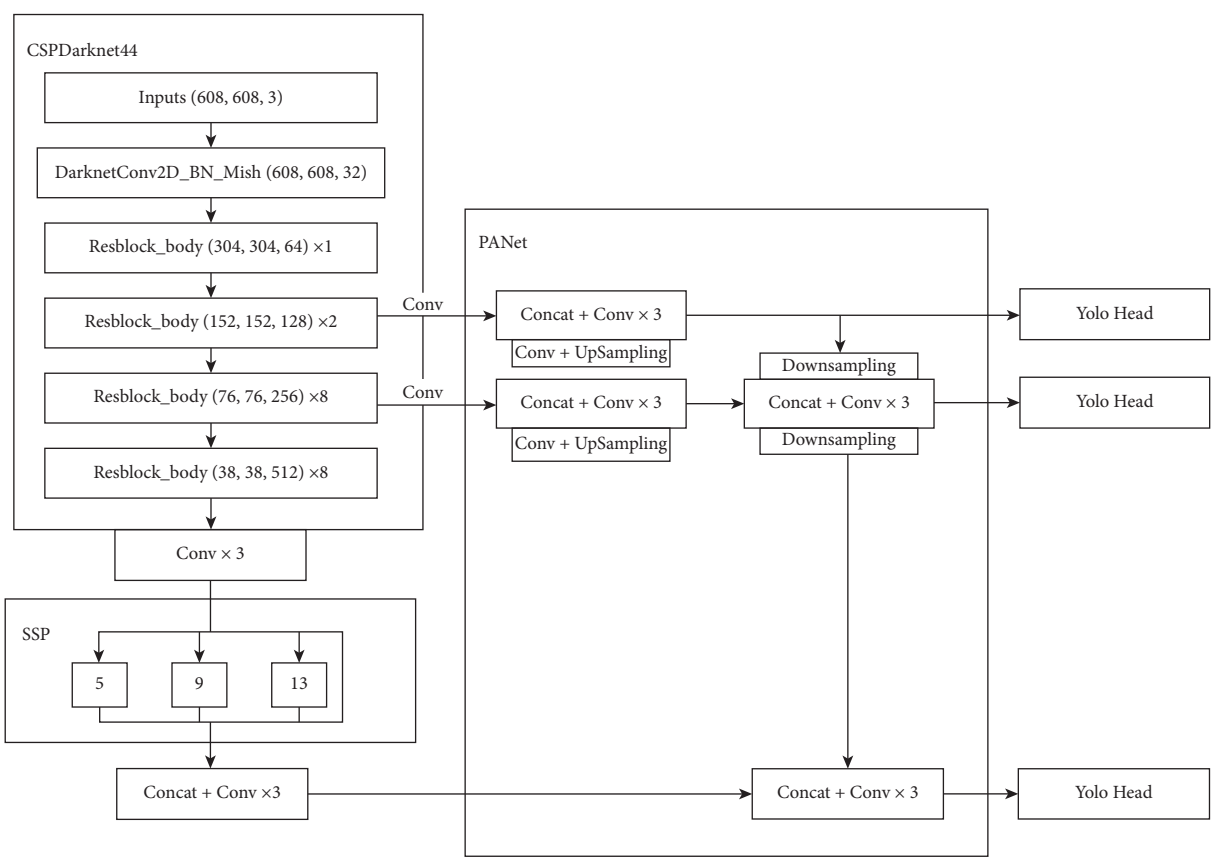

Figure 2: Lit-YOLO network structure.

anchor box and border regression are used to preset the borders to help establish the heights and widths of common targets. When making predictions, the anchor box first frames the target at possible locations. These preset borders are then fine-tuned by panning and scaling (border regression), resulting in the fine-tuned window being closer to the ground truth. The choice of anchor box size directly affects the accuracy and speed of the model.

We recalculated the anchor box size using the K-means++ clustering algorithm. K-means++ solves the problem of the K-means algorithm being greatly influenced by the initial value. Unlike the random selection of the initial cluster centers, the basic idea of the K-means++ algorithm in selecting the initial seed is to make the initial cluster centers as far away from each other as possible. The K-means++ clustering process is shown in Figure 3, and the algorithm flow is as follows:

(1) Randomly select a point in the input dataset as the initial cluster center.

(2) Calculate the distance $D(x)$ from each point x to the nearest cluster center (selected) and record it in the array ($D1$, $D2$, . . ., Dn).

(3) The next center is selected using the roulette method. The principle is that the point with larger $D(x)$ has a higher probability of being selected.

(4) Repeat steps (2) and (3) until K cluster centers are found.

(5) The standard K-means algorithm is run using the obtained K clustering centers from (4) as initial points.

The standard clustering algorithm measures the difference using the Euclidean distance. However, this method has a correspondingly larger error when the size of the box is relatively large. Consequently, the processed IOU is used as an evaluation index, as in the following equation:

$$\text{distance}(\text{box}, \text{anchor}) = 1 - \text{IOU}(\text{box}, \text{anchor}). \tag{1}$$

The predefined anchor values in the original YOLOv4 detection network were obtained by clustering on the COCO dataset. However, here, we tested a single object, only litchi. To improve the accuracy of the model, the litchi dataset was re-clustered to obtain the anchor values, as detailed in Table 2.

2.5. Cutting Detection. The resolution of the images collected in this study is 4,072 × 3,072, and the size of the litchi fruit is less than 100. Thus, a litchi fruit is an absolutely small target relative to the whole image. Usually, YOLOv4 has three detection sizes: 608 × 608, 512 × 512, and 416 × 416. The YOLOv4 network scales the input image, which leads to fewer pixels and less distinctive features for small targets. Consequently, small target detection is difficult. Therefore, when the image size is much larger than the maximum input size of the detection model, direct downsampling of the input image is not effective for detection. To achieve the detection of small targets in large scenes, the original images are cropped and inputted separately into the model for detection based on the improved model. Then, the detection results are mapped back to the original image.

Figure 4(a) shows the original input image to be processed, and the area of the image region is set to S. Because the original input image size is too large, it is difficult to detect the small-scale target objects directly as the input for the model. Here, the original image was cropped into nine copies and inputted separately into the model for detection. The image area can be expressed as follows:

$$S = \sum_{i,j=1}^{3} S_{ij}. \tag{2}$$

If the image is evenly divided into nine parts, then it appears that the complete litchi target is cut into two parts and attributed to different cropping areas. This leads to duplicate recognition counts of litchi targets or missed detection because the litchi cannot be correctly identified in Figure 4(b). To avoid the above situation and ensure that each litchi in the original image is detected completely and correctly; during the original image cropping, each cropped area should have the proper overlap with surrounding areas, as shown in Figure 4(c). The original image was cropped to an image detection region of length W and width H. The area size of each detected image after image cropping can be determined as follows:

$$S = \sum_{i,j=1}^{3} S_{ij}, \tag{3}$$

$$W_{i,j} = \frac{W}{3} + \delta_1\left(\delta_1 = \frac{W}{10}\right), \tag{4}$$

$$H_{i,j} = \frac{H}{3} + \delta_2\left(\delta_2 = \frac{H}{10}\right). \tag{5}$$

The original image area can be determined as follows:

$$S = \sum_{i,j=1}^{3}\left(W_{i,j} \times H_{i,j}\right) - \sum_{i,j=1}^{3} S_{i,j}. \tag{6}$$

Some duplicate images inevitably occur in the overlapping regions. Therefore, the NMS algorithm was used to filter the images. Finally, the results of the cropped images were combined together and mapped back to the original image position to produce the final detection results. The process of filtering duplicate boxes is as follows:

(1) Sort all the prediction boxes in order of probability from highest to lowest

(2) Eliminate prediction frames with probabilities of less than a set threshold (confidence)

(3) Calculate the intersection ratio of the highest probability box m_1 and other boxes m_2, respectively, as follows: $\text{IoU} = \text{area}(m_1) \cap \text{area}(m_2)/\text{area}(m_1) \cup \text{area}(m_2)$

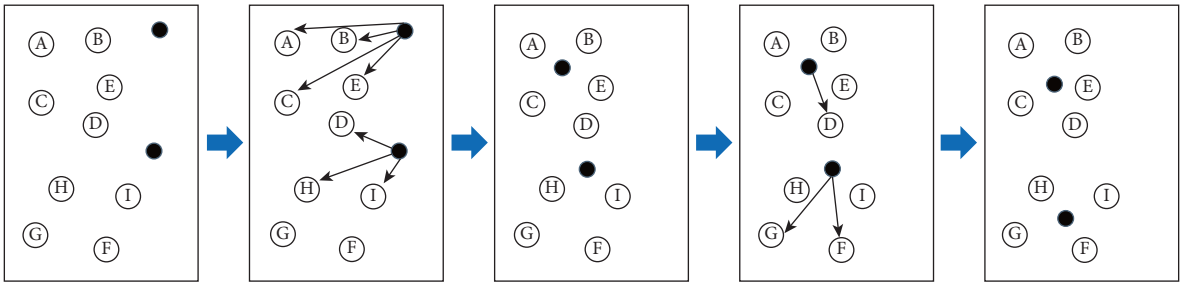

Figure 3: Standard K-means clustering process.

Table 2: The anchor values for the litchi dataset.

Feature map	Original values	Update values
Small	(12,16) (19,36) (40,28)	(5,7) (6,10) (8,12)
Medium	(36,75) (76,55) (72,146)	(10,16) (12,20) (16,24)
Large	(142,110) (192,243) (459,401)	(20,32) (32,48) (54,80)

(4) Eliminate boxes with intersection-to-merge ratio (IoU) greater than the set threshold because the overlap is too great

(5) Repeat steps (3) and (1) until all the boxes are filtered

3. Results and Discussion

3.1. Experimental Environment. The experiments were conducted using the Darknet deep learning framework. The Lit-YOLO model was compiled and tested in Python 3.6. The model was trained on a 64 bit Ubuntu 16.04 machine with an Intel i7-10700k CPU, 16G RAM, and an NVIDIA RTX 2080S GPU.

3.2. Evaluation Indicators. In general, the most basic metrics to judge the quality of a model are Precision, Recall, F1 value, and mean average precision. Calculating these indicators includes four parameters, positive sample and positive result (True Positive, TP), negative sample and negative result (True Negative, TN), negative sample and positive result (False Positive, FP), and positive sample and negative result (False Negative, FN).

Precision:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{7}$$

Recall:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{8}$$

F1 score:

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} \times \text{recall}}. \tag{9}$$

3.3. Results' Analysis. The models were tested with three input sizes of 416×416, 512×512, and 608×608, and the P-R graphs were plotted based on the test results, as shown in Figure 5.

The detection of the same model was improved when using a larger input size. For the same model, the 608×608 input size produced the best result, and the 416×416 input size produced the worst results. Moreover, the improved Lit-YOLO model shows greater accuracy, recall, and average precision values than the original YOLOv4 and Fast R-CNN algorithm. The test results are shown in Table 3.

The purpose of the intelligent monitoring of litchi orchards is to perform the statistical determination of the number of single litchi fruit in large scenarios. Therefore, the number of correctly identified litchi and the number of incorrectly identified litchi are used as the model accuracy evaluation criteria. The error identification includes false detection, missed detection, frames with multiple fruit, and a fruit in multiple frames. No litchi mark in the box indicates a false detection. Litchi that is not boxed out is recorded as a missed detection. If there are multiple litchi in a box, then only mark one is correct. If a litchi is contained in multiple boxes, then only one "correct" is marked. Because the number of litchi fruit is large, 20 images were randomly selected in the test set. The number of fruit was counted manually as 1,732. The statistical results of the recognition models presented here are shown in Table 4. There were 1,584 correctly recognized litchi, and the correct recognition rate was 91.45%. The number of incorrectly identified litchi was 148, with a recognition error rate of 8.55%.

To test the generalization ability of the model for different densities of litchi, three different scenes of sparse (small), normal (medium), and dense (large) litchi were selected for recognition detection. The actual detection results are shown in Figure 6–8.

In the images taken in a natural environment, the number and density of litchi differ greatly. It is easier to identify the litchi that are larger and clearly imaged. However, for images containing a large number of litchi, the detection is more difficult because the branches and leaves are obscured and the fruit are stuck together. The test set was divided into two gradients according to the number of fruit in the image. The gradients were less than 100 and more than 100. The correct rates of the two recognition models were compared separately. The image detection results are shown in Table 5.

As the number of fruit in the image increased, there were more cases of fruit blocking each other and sticking together.

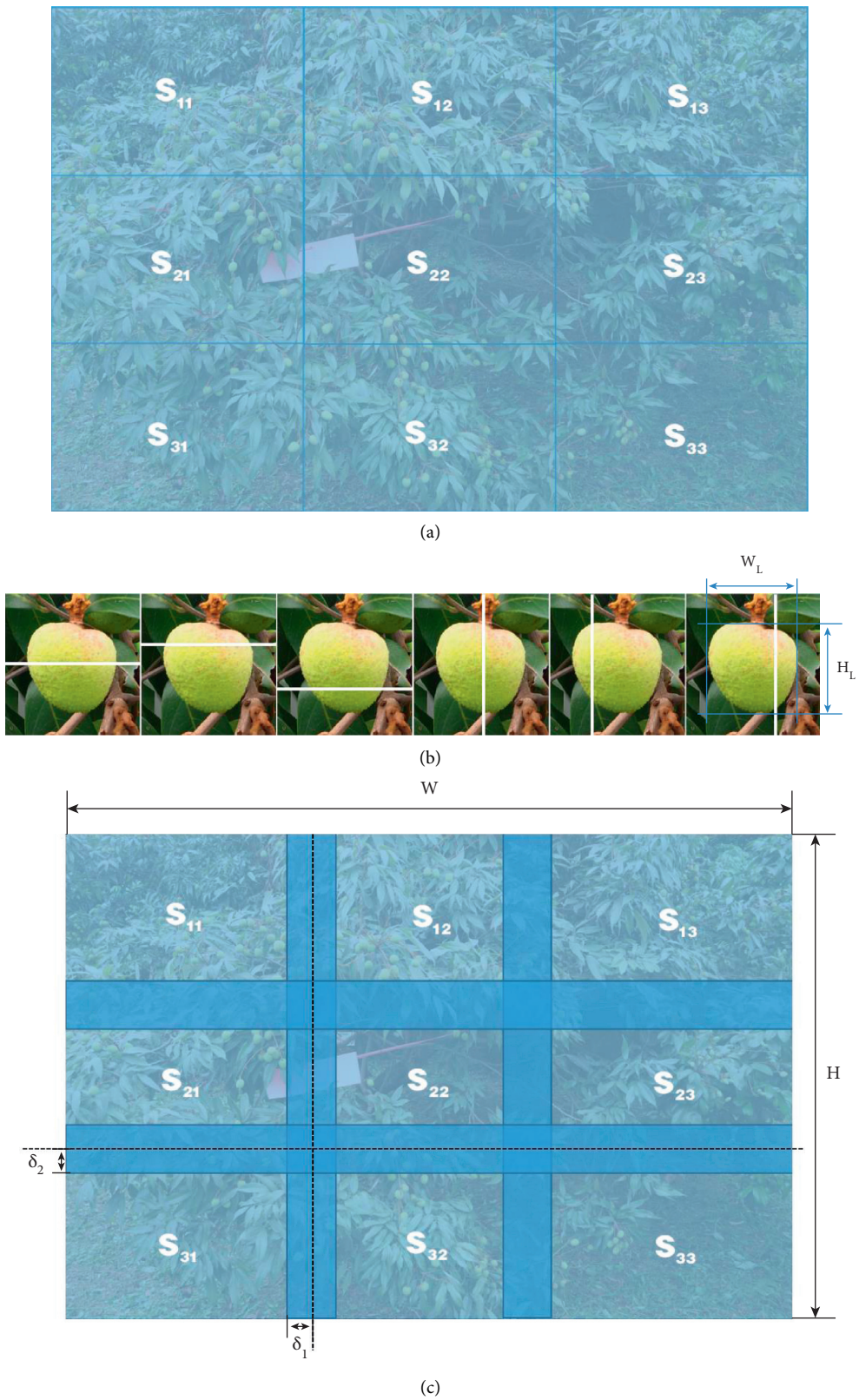

Figure 4: Image cropping diagram.

Additionally, the more fruit contained in the image, the more complex the background. The litchi in the image were also smaller and blurrier, leading to the features not being easily extracted. In dense scenes, the correct recognition rate of the original YOLOv4 model was greatly reduced. However, the correct recognition rate of the proposed improved

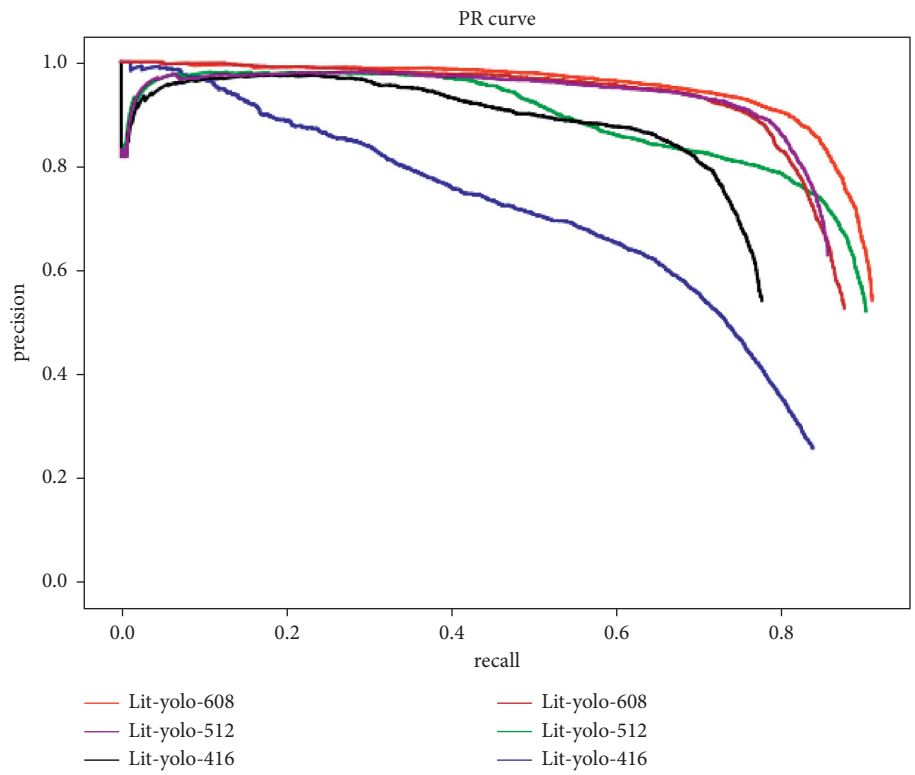

Figure 5: P-R curve graph.

Table 3: Model test results.

Model	Precision (%)	Recall (%)	F1	mAP (%)
Lit-YOLO	89.29	82.47	0.85	85.45
Yolov4	76.25	81.56	0.79	80.31
Fast R-CNN	83.34	72.58	0.77	67.29

Table 4: Model image detection comparisons.

Model	Number of fruit	Number of correctly identified	Correctness rate (%)
Lit-YOLO	1732	1584	91.45
Yolov4	1732	1460	84.30

Figure 6: YOLOv4 and Lit-YOLO detection results in sparse scenes.

Figure 7: YOLOv4 and Lit-YOLO detection results in normal scenes.

Figure 8: YOLOv4 and Lit-YOLO detection results in dense scenes.

Table 5: Comparison of detection results at different fruit densities.

Density of fruit in the image	Number of images	Number of fruits	Lit-YOLO		YOLOv4	
			Identification number	Identification rate (%)	Identification number	Identification rate (%)
Sparse	15	522	478	91.57	463	88.69
Normal	12	901	834	92.56	802	89.01
Dense	8	1148	1052	91.64	944	82.23

model does not change much at any litchi density. The correct rate was 2.88% greater than the YOLOv4 prediction in sparse scenes. The correct rate was 3.55% greater than the YOLOv4 prediction in normal scenes. In dense scenes, the correct rate was 9.41% greater than the YOLOv4 prediction. Thus, the improved model was effective for the detection of a large number of densely distributed targets in a large scene. It also met the demand for real-time detection of litchi fruit in litchi orchards.

4. Conclusions

Intelligent monitoring of litchi orchards based on the IoT and artificial intelligence technology can be used for real-time data collection from litchi fruit trees and of fruit growth, to carry out intelligent planting decisions to increase litchi orchard production by eliminating traditional orchard operations and management practices. These are important developmental direction for litchi orchards. To improve the real-time recognition and detection of litchi fruit, this study proposed an improved Lit-YOLO model based on the deep learning YOLOv4 model. The K-means++ clustering algorithm was first used to obtain the anchor data suitable for the litchi dataset. A 32-fold downsampled network was removed from the original network, and a 4-fold downsampled feature map was produced for predicting small targets. Sliding windows with overlaps were cropped from the original map before being inputted into the model, and they were inputted separately into the model for prediction. Finally, the overlapping prediction frames were filtered using the NMS algorithm. With 400 scene images (320 images in the training set and 80 images in the test set)

detected, the improved Lit-YOLO model has a recognition rate of over 91% in both scenes, having more and less numbers of fruit. The model was robust for the detection of densely distributed small targets in large scenes. The recognition of obscured litchi was also more accurate than the original model. Accurate and reliable fruit recognition may provide local information for robotic picking. It also allows for yield prediction and proper marketing planning. The improved methodology of the new model may be applied to the detection of other small fruit with dense distributions in a natural environment.

Data Availability

The litchi detection data used to support the findings of this study have not been made available. Currently, no data were uploaded, and the manuscript data will be updated after acceptance.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by grants from the Guangdong Agricultural Research Project and the Agricultural Technology Promotion Project. The project name is litchi automatic picking equipment key components development and testing (no. F21137) and the project by the South China Agricultural University, in conjunction with Guangdong RuoBo Intelligent Co. Ltd. The authors thank Zhen Long Shan Ding village litchi professional cooperative four units to carry out research and test verification.

References

[1] H. B. Chen and H. B. Huang, "China litchi industry: development, achievements and problems," *Acta Horticulturae*, vol. 558, pp. 31–39, 2001.

[2] H. B. Chen and X. M. Huang, "Overview of litchi production in the world with specific reference to China," *Acta Horticulturae*, vol. 1029, pp. 25–33, 2014.

[3] C. Zhang, J. Valente, L. Kooistra, L. Guo, and W. Wang, "Orchard management with small unmanned aerial vehicles: a survey of sensing and analysis approaches," *Precision Agriculture*, pp. 1–46, 2021.

[4] S. Park, A. Nolan, D. Ryu et al., "Estimation of crop water stress in a nectarine orchard using high- resolution imagery from unmanned aerial vehicle (UAV)," in *Proceedings of the 21st International Congress on Modelling and Simulation*, Gold Coast, Australia, November 2015.

[5] S. Khan, M. Tufail, M. T. Khan, Z. A. Khan, and S. Anwar, "Deep-learning-based spraying area recognition system for unmanned-aerial-vehicle-based sprayers," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 29, no. 1, pp. 241–256, 2021.

[6] W. Sun, Z. Wang, J. Ding, W. Lu, and Y. Sun, "Remote measurement of apple orchard canopy information using unmanned aerial vehicle photogrammetry," *Agronomy*, vol. 9, no. 11, p. 774, 2019.

[7] J. Abdulridha, O. Batuman, and Y. Ampatzidis, "UAV-based remote sensing technique to detect citrus canker disease utilizing hyperspectral imaging and machine learning," *Remote Sensing*, vol. 11, no. 11, p. 1373, 2019.

[8] G. Caruso, Z. P. J. Tejada, V. G. Dugo et al., "High-resolution imagery acquired from an unmanned platform to estimate biophysical and geometrical parameters of olive trees under different irrigation regimes," *Plos One*, vol. 14, no. 1, Article ID e0210804, 2019.

[9] S. F. Di Gennaro, P. Toscano, P. Cinat, A. Berton, and A. Matese, "A low-cost and unsupervised image recognition methodology for yield estimation in a vineyard," *Frontiers of Plant Science*, vol. 10, 2019.

[10] S. Sabzi, A. Y. Gilandeh, G. G. Mateos, A. R. Canales, J. M. Martínez, and J. Arribas, "An automatic non-destructive method for the classification of the ripeness stage of red delicious apples in orchards using aerial video," *Agronomy Journal*, vol. 9, no. 2, 2019.

[11] A. Taofik, N. Ismail, Y. A. Gerhana, K. Komarujaman, and M. A. Ramdhani, "Design of smart system to detect ripeness of tomato and chili with new approach in data acquisition," *IOP Conference Series: Materials Science and Engineering*, vol. 288, Article ID 012018, 2018.

[12] J. Zhuang, C. Hou, Y. Tang et al., "Computer vision-based localisation of picking points for automatic litchi harvesting applications towards natural scenarios," *Biosystems Engineering*, vol. 187, pp. 1–20, 2019.

[13] 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.

[14] P. J. Ramos, F. A. Prieto, E. C. Montoya, and C. E. Oliveros, "Automatic fruit count on coffee branches using computer vision," *Computers and Electronics in Agriculture*, vol. 137, pp. 9–22, 2017.

[15] A. Aquino, B. Millan, M.-P. Diago, and J. Tardaguila, "Automated early yield prediction in vineyards from on-the-go image acquisition," *Computers and Electronics in Agriculture*, vol. 144, pp. 26–36, 2018.

[16] A. B. Payne, K. B. Walsh, P. P. Subedi, and D. Jarvis, "Estimation of mango crop yield using image analysis - segmentation method," *Computers and Electronics in Agriculture*, vol. 91, pp. 57–64, 2013.

[17] M. Chen, Y. Tang, X. Zou, Z. Huang, H. Zhou, and S. Chen, "3D global mapping of large-scale unstructured orchard integrating eye-in-hand stereo vision and SLAM," *Computers and Electronics in Agriculture*, vol. 187, Article ID 106237, 2021.

[18] H. Kang and C. Chen, "Fruit detection and segmentation for apple harvesting using visual sensor in orchards," *Sensors*, vol. 19, no. 20, p. 4599, 2019.

[19] 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, p. 1016, 2020.

[20] 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.

[21] 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, Article ID 105508, 2020.

[22] H. A. M. Williams, M. H. Jones, M. Nejati et al., "Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms," *Biosystems Engineering*, vol. 181, pp. 140–156, 2019.

[23] 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.

[24] J. J. Zhuang, S. M. Luo, C. J. Hou, Y. Tang, Y. He, and X. Y. Xue, "Detection of orchard citrus fruits using a monocular machine vision-based method for automatic fruit picking applications," *Computers and Electronics in Agriculture*, vol. 152, pp. 64–73, 2018.

[25] M. Faisal, M. Alsulaiman, M. Arafah, and M. A. Mekhtiche, "IHDS: intelligent harvesting decision system for date fruit based on maturity stage using deep learning and computer vision," *IEEE Access*, vol. 8, pp. 167985–167997, 2020.

[26] Q. Lu, J. Cai, J. Zhao, W. Feng, and T. MingJie, "Real-time recognition of citrus on trees in natural scene," *Nongye Jixie Xuebao/Transactions of the Chinese Society of Agricultural Machinery*, vol. 41, no. 2, pp. 185–188+170, 2010.

[27] Y. Liu, B. Chen, and J. Qiao, "Development of a machine vision algorithm for recognition of peach fruit in a natural scene," *Transactions of the Asabe*, vol. 54, no. 2, pp. 695–702, 2011.

[28] L. Fu, J. Duan, X. Zou et al., "Banana detection based on color and texture features in the natural environment," *Computers and Electronics in Agriculture*, vol. 167, Article ID 105057, 2019.

[29] L. Yu, J. Xiong, X. Fang et al., "A litchi fruit recognition method in a natural environment using RGB-D images," *Biosystems Engineering*, vol. 204, no. 1, pp. 50–63, 2021.

[30] R. Girshick, "Fast R-CNN," 2015, https://arxiv.org/abs/1504.08083.

[31] 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, pp. 1137–1149, 2017.

[32] X. Zhang, Q. Gao, D. Pan, P. C. Cao, and D. H. Huang, "Research on spatial positioning system of fruits to be picked in field based on binocular vision and SSD model," *Journal of Physics: Conference Series*, vol. 1748, no. 4, Article ID 042011, 2021.

[33] 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.

[34] H. Yang, L. Chen, M. Chen et al., "Tender tea shoots recognition and positioning for picking robot using improved YOLO-V3 model," *IEEE Access*, vol. 7, pp. 180998–181011, 2019.

[35] R. Shi, T. Li, and Y. Yamaguchi, "An attribution-based pruning method for real-time mango detection with YOLO network," *Computers and Electronics in Agriculture*, vol. 169, Article ID 105214, 2020.