Litchi detection in the field using an improved YOLOv3 model

Hongxing Peng[1], Chao Xue[1], Yuanyuan Shao[2], Keyin Chen[3], Huanai Liu[4*], Juntao Xiong[1*], Hu Chen[1], Zongmei Gao[5], Zhengang Yang[1]

(1. *College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China*;
2. *College of Mechanical and electronic engineering, Shandong Agricultural University, Tai'an 271018, Shandong, China*;
3. *School of Electronic and Information Engineering, Jiaying University, Meizhou 514015, Guangdong, China*;
4. *School of Chemistry and Chemical Engineering, South China Technology of University, Guangzhou 510641, China*;
5. *Center for Precision and Automated Agricultural Systems, Department of Biological Systems Engineering, Washington State University, Prosser, WA 99350, USA*)

Abstract: Due to the illumination, complex background, and occlusion of the litchi fruits, the accurate detection of litchi in the field is extremely challenging. In order to solve the problem of the low recognition rate of litchi-picking robots in field conditions, this study was inspired by the ideas of ResNet and dense convolution and proposed an improved feature-extraction network model named "YOLOv3_Litchi", combining dense connections and residuals for the detection of litchis. Firstly, based on the traditional YOLOv3 deep convolution neural network and regression detection, the idea of residuals was to be put into the feature-extraction network to effectively avoid the problem of decreasing detection accuracy due to the excessive depths of the network layers. Secondly, under the premise of a good receptive field and high detection accuracy, the large convolution kernel was replaced by a small convolution kernel in the shallow layer of the network, thereby effectively reducing the model parameters. Finally, the idea of feature pyramid was used to design the network to identify the small target litchi to ensure that the shallow features were not lost and simultaneously reduced the model parameters. Experimental results show that the improved YOLOv3_Litchi model achieved better results than the classic YOLOv3_DarkNet-53 model and the YOLOv3_Tiny model. The mean average precision (mAP) score was 97.07%, which was higher than the 95.18% mAP of the YOLOv3_DarkNet-53 model and the 94.48% mAP of the YOLOv3_Tiny model. The frame frequency was 58 fps, which was higher than 29 fps of the YOLOv3_DarkNet-53 model. Compared with the classic Faster R-CNN model with the feature-extraction network VGG16, the mAP was increased by 1%, and the FPS advantage was obvious. Compared with the classic single shot multibox detector (SSD) model, both the accuracy and the running efficiency were improved. The results show that the improved YOLOv3_Litchi model had stronger robustness, higher detection accuracy, and less computational complexity for the identification of litchi in the field conditions, which should be helpful for litchi orchard precision management.

Keywords: deep learning, residual network, dense connection, feature pyramid network
DOI: 10.25165/j.ijabe.20221502.6541

Citation: Peng H X, Xue C, Shao Y Y, Chen K Y, Liu H N, Xiong J T, et al. Litchi detection in the field using an improved YOLOv3 model. Int J Agric & Biol Eng, 2022; 15(2): 211–220.

1 Introduction

The litchi is a characteristic fruit in South China. However, the litchi has a short harvesting period and always ripens in May and June. Litchi fruits may become overripe or cracked if not picked in time, resulting in direct economic losses. On the one hand, litchi picking is traditionally laborious. However, manual picking brings many unavoidable problems, such as picking at night, on rainy days, or at high temperatures, which bring huge labor and economic costs to the owners of the litchi orchard. On the other hand, litchi trees are usually planted in hilly areas with uneven terrain and the distribution of fruit trees is very scattered. Many unfavorable factors bring great challenges to the standardization and mechanization of litchi production, especially picking.

Therefore, an automated litchi picking robot is needed to resolve the above issues. However, if the robot can pick litchi efficiently and accurately, litchi fruit detection and recognition are the prerequisites.

In recent years, there have been many studies on fruit identification using traditional machine vision algorithms across the world. Wei et al.[1] proposed an improved OTSU threshold algorithm using new features in the OHTA color space, which improved the ability to pick robots to identify the fruit targets in complex agricultural backgrounds. Zhuang et al.[2] used a block-based local homomorphic filtering algorithm to ensure that

Received date: 2021-02-22 **Accepted date:** 2021-08-12
Biographies: Hongxing Peng, PhD, Associate Professor, research interests: machine vision, agricultural robot, artificial intelligence, Email: xyphx@scau.edu.cn; **Chao Xue**, MS, Engineer, research interests: machine vision, Email: 18764815162@163.com; **Yuanyuan Shao**, PhD, Associate Professor, research interests: smart agriculture, Email: syy007@sdau.edu.cn; **Keyin Chen**, PhD, Lecturer, research interests: machine vision, agricultural robot, Email: chenkeyin10@126.com; **Hu Chen**, MS candidate, research interests: machine vision, Email: robo_ch_offical@163.com; **Zongmei Gao**, PhD, Postdoctor, research interests: smart agriculture, Email: zongmei.gao@wsu.edu; **Zhengang Yang**, PhD, Associate Professor, research interests: agricultural robot, Email: yzg@scau.edu.cn.

***Corresponding author: Huanai Liu**, MS, Senior Experimentalist, research interests: calculational chemistry. School of Chemistry and Chemical Engineering, South China Technology of University, Guangzhou 510641, China. Tel: +86-15918569718, Email: liuhn@scut.edu.cn; **Juntao Xiong**, PhD, Associate Professor, research interests: machine vision, agricultural robot. College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China. Tel: +86-13560164695, Email: xiongjt2340@163.com.

only local blocks having a non-uniform illumination distribution were filtered, and threshold segmentation was better performed by adaptively enhanced RG chromaticity mapping. In order to improve the recognition ability and perception ability of robots in three-dimensional space, Tao et al.[3] proposed a method for apple recognition, which used point cloud information to extract color features and three-dimensional geometric features, and then the genetic algorithm was used to optimize the parameters of the SVM classifier. Also, there are some methods for determining the degree of maturity[4].

With the development of machine vision algorithms and the continuous improvement of parallel computing capabilities, deep learning as a popular technology has obvious advantages in the field of computer vision. As a deep learning method, convolutional neural networks have advantages in processing graphics[5]. The application of deep learning in computer vision is mainly divided into classification, detection, and semantic segmentation. This study was mainly engaged in research of detection. Feature-extraction networks used in target detection came from the classification networks, and the feature maps with high abstraction were obtained through the convolutional neural networks, after which the abstract information could be obtained. Google Labs proposed a breadth-based feature-extraction network and achieved good results[6]. At the same time, various deep learning algorithms have also been applied to the fruit recognition process including litchi recognition. Peng et al.[7] proposed a method that combines the DeepLabV3+ semantic segmentation model with the Xception depth separable convolution feature to detect litchi branches, and obtained good recognition results. Kang et al.[8] proposed a computational efficient light-weight one-stage instance segmentation network, Mobile-DasNet, to perform fruit detection and instance segmentation on sensory data. A deeper feature-extraction network has also emerged[9]. Sa et al.[10] proposed a novel multi-modal information fusion Faster-RCNN model using color (RGB) image and near-infrared (NIR) image information, which improved the F1 value of sweet pepper detection from 0.807 to 0.838. Bargoti et al.[11] proposed an image processing framework for fruit detection and counting, such as feature-learning algorithms including multi-scale multilayer perceptron (MLP) and convolutional neural network (CNN) algorithms. Their results show the F1 value reached 0.861. There have been some studies using convolutional neural networks for fruit classification[12] and mango detection[13].

Currently, target detection based on deep learning is mainly divided into two categories. One is the detection based on a region proposal network (RPN), such as RCNN, Fast-RCNN, and Faster R-CNN[14]. The other one is detection based on regression over the entire image to achieve synchronous prediction of the target classification and positioning, such as in YOLO[15] and SSD[16]. In recent years, the efficiency and accuracy of YOLO in the application of target detection have been continuously improved. It is gradually applied to the detection of fruit. Tian et al.[17] proposed an improved YOLOv3 model for detecting apples during different growth stages using the same model, which used the Densenet method to process the low-resolution feature layer in the YOLOv3 network. Zhao et al.[18] analyzed the application of the YOLOv2 model to the detection of healthy and diseased tomatoes and found that YOLOv2 can be effectively applied to the detection of healthy and diseased tomatoes. Liu et al.[19] replaced the traditional R-Bbox with a proposed C-Bbox, for matching the tomato shape and providing more precise IoU for the NMS process, and reducing prediction coordinates to better recognize the tomatoes.

However, although there have been some successful cases of applying the YOLO model to fruit detection, it has not been widely used. In this study, a feature extraction network model named YOLOv3_Litchi was proposed based on the idea of residual structure, dense convolution, and feature pyramid. Specifically, this study was conducted to the identification of litchis in the field. Based on YOLOv3 network in deep learning, a new feature extraction network YOLOv3_Litchi was proposed that combined residual and dense connection ideas with YOLOv3 improvements. For evaluating the robustness and detection accuracy of YOLOv3_Litchi, the performance of such model was compared with the traditional YOLOv3 model and other improved YOLOv3 models using the Litchi images collected in field.

2 YOLO deep learning model based on regression

The regression-based YOLO detection model changes the detection structure of the regional proposed network. YOLO uses the global regression idea to divide an entire image into squares of S×S for prediction (In the experiment of this study, the size of the input image was set to 416×416, the same as the original YOLOv3.). Each square is analyzed to predict whether the target is in its center. The method of predicting the candidate box, confidence, and category probability of all the cells will solve the detection problem at one time, as shown in Figure 1.

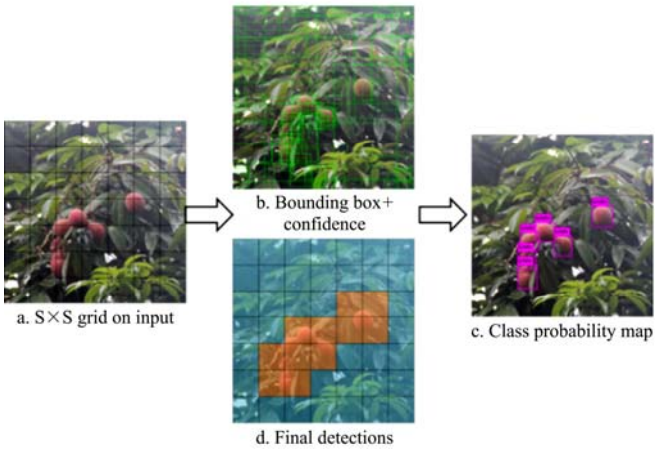
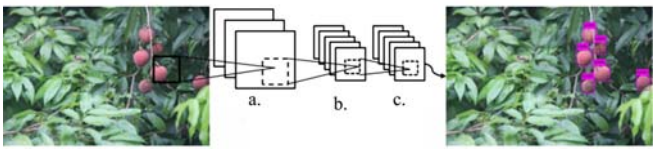


Figure 1 Schematic of YOLO square detection process

The regression-based detection idea is adopted because the Faster R-CNN series using the region proposal network (RPN) is slow in detection and difficult to train. Its purpose is to perform the high-level abstraction of images through convolutional neural networks and use regression prediction to complete the detection of targets. The structure is simple and easy to implement. It can be trained end-to-end and has a faster detection speed than that of the Faster R-CNN series, thus, it has strong universal applicability in the industry.

Although a larger input image can save more information, it can also greatly increase the amount of calculation for training the network and affect the overall performance of the network. Therefore, the input image normally needs to be resized. In this study, the scaled image was sent to the convolutional neural network for high-level feature extraction. After feature extraction, the non-maximum suppression (NMS)[20] algorithm was used to reduce the excess bounding boxes. Finally, detection of the target was conducted. Figure 2 and Figure 3 show the example of detection network and detection results, respectively.



Note: a. Resize image; b. Convolutional neural network; c. Non-maximum suppression.

Figure 2 YOLO detection network of litchi

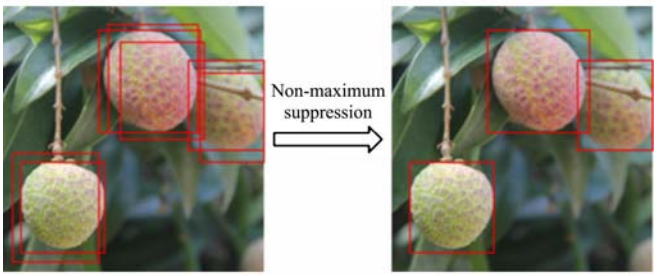


Figure 3 Detection results with non-maximum suppression of litchi

After the development of several versions of the YOLO network, the latest version of YOLOv3 has incorporated the advantages of many target detection networks. The input image of YOLOv3 is 416×416, the feature map obtained after convolution is 13×13, and a prediction is made with the 13×13 image. Up-sampling is performed to obtain a larger feature map and a shallow feature map for fusion, and then the prediction is performed. The classic YOLOv3 combines three scales, 13×13, 26×26, and 52×52, thus making a big breakthrough in small target detection.

3 Model selection, improvement, and characterization

The classic YOLOv3 detection network uses DarkNet-53 as a feature-extraction network. It contains 53 convolution layers, which are powerful but have too many layers and a large computational cost. This section would conduct optimization based on these limitations.

3.1 Residual network

Based on experience, the deep network can extract more abstract and high-dimensional features to obtain better detection results, but the deep feature network can reduce the accuracy of the training results. The deep residual network (ResNet) was designed to solve such problems[21]. In theory, without the influence of multiple layers, the network with residual structure is always in the optimal state.

The residual network played a very important role in the feature-extraction network due to its simple structure of units and low computational cost. The residual network based on Shortcut Connection, as shown in Figure 4, was composed of several residual blocks in series, effectively solving the problem of gradient disappearance and gradient explosion of deep network training, making deep network training possible.

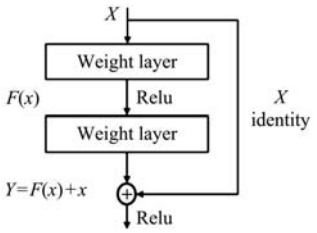
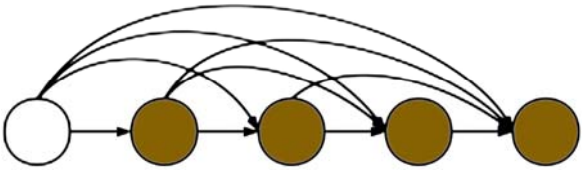


Figure 4 Shortcut connection of residual blocks

3.2 Dense convolution block

In order to ensure the maximum information to be transferred between the layers, the problems of deep network gradient disappearance and the transmission of features need to be solved. This study used dense blocks[22]. As shown in Figure 5, Dense Block was a module with many layers, each of them has the same size feature map, and the layers were closely connected to each other.



Note: The circles represent the dense blocks in groups of five, the hollow circles represent the first dense block, and the solid circles represent the four subsequent dense blocks.

Figure 5 Dense block composite structure used in this study

In traditional neural networks, deep convolutions receive more abstract information, and shallow information is filtered during the convolution and pooling operations. Through the splicing with the results of the front layer, the dense convolution block could effectively alleviate the problem of shallow information loss and gradient disappearance during the transfer process.

Dense convolution connection used Equation (1) to merge the convolution results of each layer. Compared with the ResNet, the dense convolution connection spliced the convolution results of the upper layer and retained the feature mapping to a greater extent.

$$x_l = H_l([x_0, x_1, \ldots, x_{l-1}]) \quad (1)$$

where, H_l represents the nonlinear transformation function of each layer and is defined as the combination function of three operations (namely BN, ReLU, and convolution), where l represents the layer. The output of layer L is represented by x_l.

As shown in Figure 6, the entire classification network was composed of multiple densely connected convolutional blocks. Since each dense convolutional block contained the lowest-level feature information, therefore, deeper network pooling layers would not cause all the shallow information to be lost, and it would have a better performance in the deep abstraction of details. Especially for litchi image data sets, the litchi that needed to be detected usually occupied a small part of the entire image (whole litchi), moreover, there were a lot of noise factors such as branches and leaves around litchi.

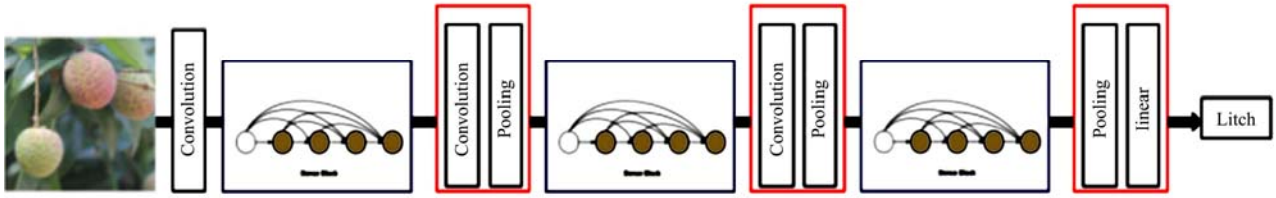


Figure 6 DenseNet classification network for litchi classification

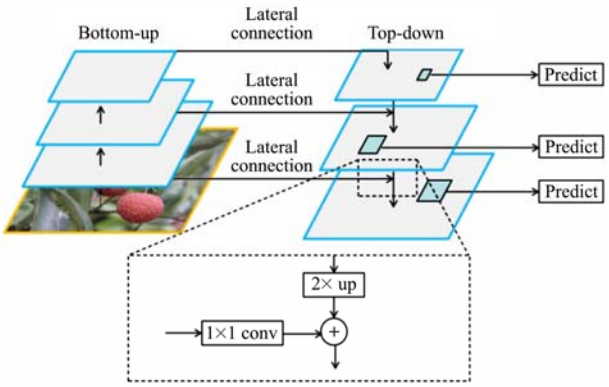
By reasonably using the DenseNet structure in the network of this study, the loss of the characteristics of small targets was able to avoid such as litchi in the deep network structure.

3.3 Feature pyramid for litchi

After the training samples were input into the network, they needed to pass through the convolutional layer and the pooling

layer of the feature extraction network, which would make the network discard the shallow features and reduce the number of the network parameters. This would make the final features extracted by the network only contain high-level abstract features of large targets, while the features of small targets were discarded because they consist of fewer pixels and the features are not obvious enough. Therefore, this study combined the idea of feature pyramid[23] with the YOLOv3 network model to solve the above problems.

As shown in Figure 7, the Feature Pyramid Network (FPN) could be divided into three main structures: bottom-up pathway, lateral connection, and top-down pathway. The bottom-up pathway was the feedforward calculation of the trunk ConvNet, which calculated the feature hierarchy consisting of several proportional feature maps with a proportional step size of 2. The top-down pathway produced higher resolution features by upsampling spatially coarser but semantically stronger feature graphs from a higher pyramid level. These features were then enhanced by the features of the bottom-up pathway through lateral connections. Each lateral connection merged feature maps of the same spatial size from the bottom-up path and the top-down path. Finally, the feature images obtained from different branches were merged by an addition operation in order to obtain the feature images with more information.



Note: "+" indicates that horizontal connections and top-down channels are merged by addition.

Figure 7 Feature pyramid network structure diagram

Due to the small size of litchi, it took up fewer pixels in high-resolution images. If a single-scale feature map was fed for training, the feature information of small targets such as litchi in the convolved feature map was easily lost. Therefore, in the YOLOv3_Litchi network structure, using the idea of FPN, a bottom-up downsampling process was performed on high-resolution litchi images and used top-down channels to upsample the top-level feature map. Finally, to avoid the influence of noising data such as branches and leaves of high-resolution images, the lateral connections were used to output multi-scale feature maps for the identification and classification of litchi in the last three layers. From the experimental data in the experiment and results, after adding the improved FPN structure, YOLOv3_Litchi had produced excellent performances for the recognition of litchi with small volume.

3.4 Small convolution replaces large convolution

The convolution operation has regularization effects by reducing the training parameters and improving the training efficiency. As the network grows larger, the detection accuracy also increases, and the parameters in the network continue to increase. Reducing the size of the convolution kernel can effectively reduce the number of parameters while ensuring the same accuracy. It is also one of the future trends. Ye et al.[24] used small convolution kernels to reduce the network parameters of the capsule network and optimized the number and quality of capsules in the capsule network on the premise of ensuring the accuracy of the network, improving the computational efficiency. Tek et al.[25] proposed a method for learning the size of the convolution kernel to provide a variable size kernel in a single layer, and optimized the overall network model from the perspective of adjusting the size of the convolution kernel.

In this study, 1×1 small convolution kernel was used in the shallow layer of the network to replace the large convolution, which reduced the network model parameters and improved the model detection speed on the premise of ensuring the overall network identification accuracy.

3.5 YOLOv3_Litchi architecture

This study used the ideas mentioned above to improve the feature extraction network, and an improved YOLOv3 network structure named "YOLOv3_Litchi" was proposed.

The YOLOv3_Litchi network structure is shown in Figure 8. In this structure:

CBL was a single convolutional block that was locally normalized. It was the smallest part of the YOLOv3 network structure, and it consisted of three parts: convolution, Batch Normalization, and Leaky_Relu activation functions.

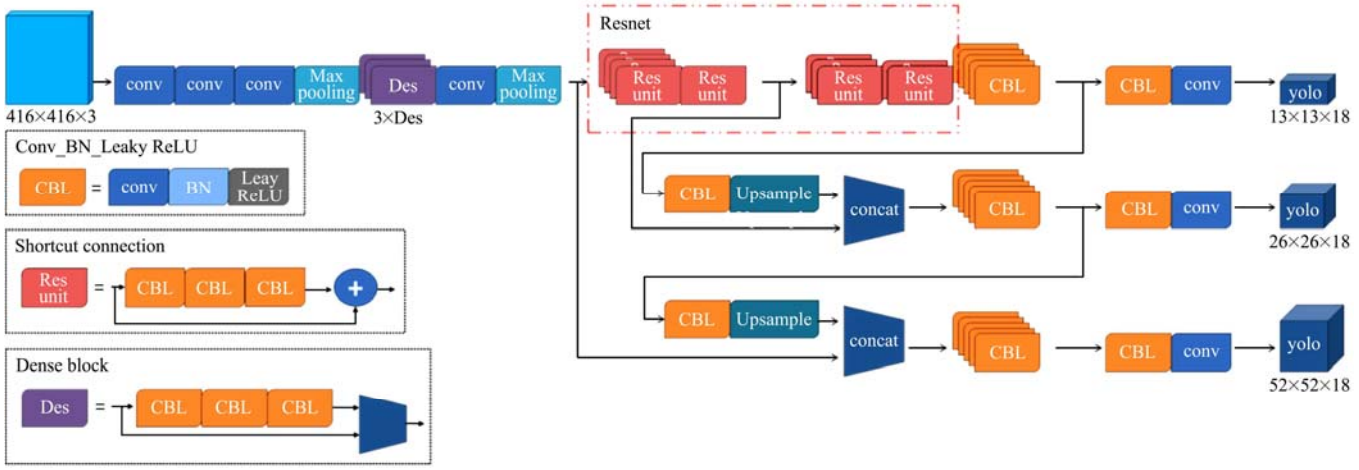
Res unit was a single residual unit, and each residual unit was subjected to Leaky_Relu function excitation after convolution. Then, the convolved result was accumulated with the original identity map, and the linear excitation was used to obtain the unified result.

Des was a densely concatenated convolution block with three convolutions. The convolved results were spliced with the pre-convolution feature maps so that the shallow features could be acquired at a deep level.

Figure 9 shows the specific network parameters of YOLOv3_litchi. Firstly, the number of convolution layers in ResNet50 was reduced, and the input image was enlarged to 416×416. Secondly, the idea of using multiple small convolutions to replace large convolutions was implemented. On the premise of the same receptive field and precision, three 3×3 small convolution kernels were used instead of the large 7×7 convolution kernel in the ResNet network, reducing the model parameters. Thirdly, the shallow network used dense convolutional blocks to quickly convey shallow information to deep convolution, which facilitated the combination of upsampling information with shallow features for regression prediction. Fourthly, in the process of network construction, the convolutional cores of different sizes were used to build the network and test the performance, and decided to adopt the 1×1 convolutional kernel. Since the length and width of the 1×1 convolution kernel are only 1 pixel, it is not needed to consider the relationship between pixels and surrounding pixels. It was mainly used to adjust the number of channels and multiple places in the network to facilitate cross-channel interaction and information integration. To ensure the constant size of the feature map, the nonlinear features were increased to obtain higher layer features. Finally, in the second half of the feature extraction network, the residual network was used to deeply extract the shallow information features which were extracted from the dense convolution block to obtain high-dimensional features and improve the prediction accuracy. Meanwhile, the feature extraction network performed a regression prediction after obtaining the

feature map. Using the idea of feature pyramid network, the entire network performed two up-samplings and three regression predictions. Combining the upsampling results with the shallow feature maps overcame the shortcomings of the YOLO network in predicting shallow small targets, and the recombined YOLOv3_Litchi network model produced good recognition performance for litchis in small sizes.

When extracting deep features, a densely connected convolution block and a third residual block were used to extract deep features, which used small convolution kernel and identity mapping ideas, and batch normalization was performed after each convolution of the residual block. This ensured that the training data were all at the same magnitude, and made training more stable and faster to converge. The non-saturated activation function Leaky ReLU function was used to accelerate the convergence and solve the problem of vanishing gradient.



Note: CBL is a single convolutional block that was locally normalized; Res unit is a single residual unit, and each residual unit was subjected to Leaky_Relu function excitation after convolution. Des was a densely concatenated convolution block with three convolutions; BN: Batch normalization; ReLU: Rectified linear units.

Figure 8 Network structure of YOLOv3_Litchi

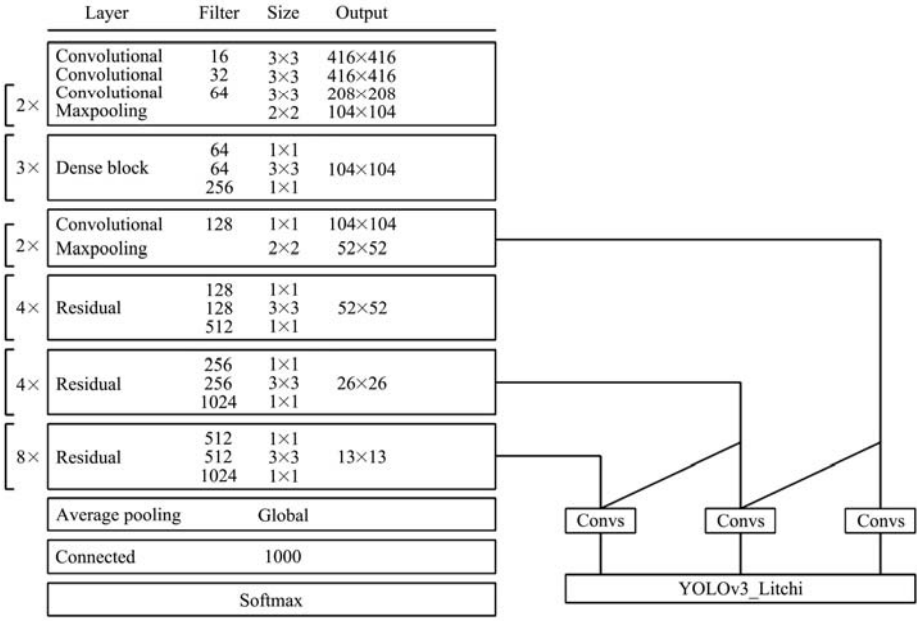


Figure 9 YOLOv3_Litchi network parameters

4 Experiment

4.1 Image data acquisition

The work of data collection was conducted in Litchi orchards in Guangzhou Green Water Fruit Farm and Guangzhou East Forest Fruit Park. The distribution of litchi trees in the two orchards was uneven. The litchi varieties included Guiwei, Zizixiao, Huaizhi, and Nuomici. The data set has been collected a total of four times. The collection dates of the litchi images in this experiment were June 29, 2017 (sunny), July 8, 2017 (cloudy to rainy), July 10, 2017 (sunny), and May 30, 2018 (sunny). In order to efficiently collect the image data under different resolutions, three kinds of camera were used for image capture. The Canon EOS 60D camera (Manufactured by Canon) was used to capture 5184 pixels×3450 pixels images, the FinePix F500EXR camera (Manufactured by FUJIFILM) was used to capture 4608 pixels×3456 pixels images, and HUAWEI smart mobile phones (Manufactured by HUAWEI TECHNOLOGIES CO., LTD) were used to capture 3968 pixels×2976 pixels images. The weather conditions included rainy, cloudy, and sunny days, and the times for image acquisition were from 8:00 to 17:00. The image data contained large differences for the convenience of strengthening the robustness and test difficulty of the detection network.

4.2 Image data pre-processing

Since a large amount of image data would result in unnecessary space and time costs for data storage and computation, based on the original data, the image was compressed into a different size. Total number of images was 4748, and the storage

space was 14 GB. After preprocessing, the image data took only 714 MB, which reduced the storage space by 20 times. The summary of image data was shown in Table 1. The processed data set was organized using the PASCAL VOC data set format, with the division principle of training: validation: test with a ratio of 5:2:3 to ensure the randomness and test reliability of images with mutually exclusive data sets.

Table 1 Image resolution and dataset information of litchi

Original image resolution (pixels×pixels)	Number of data sets				Resolution after compression (pixels×pixels)
	Training	Validation	Test	Total	
1000×1500	7	3	4	14	1000×1500
2976×3968	82	32	50	164	800×1066
3968×2976	242	96	146	484	800×600
3456×2304	215	86	130	431	800×533
5184×3450	1852	733	1070	3655	1000×666
Total	2398	950	1400	4748	--

4.3 Supervised learning annotation

This experiment used LabelImg annotation tool for supervised learning. The data was annotated in the format of the PASCAL VOC dataset. The XML file was used to store the coordinates of the upper left and lower right corners of the label target, i.e., (X_{min}, Y_{min}, X_{max}, Y_{max}), to denote the specific location of the target. In this experiment, 4748 images were used to obtain 19 907 labeled litchi targets, which met the requirements of deep learning data.

After the data annotation work was completed, the marked file was processed by converting the absolute position to the relative position adopted by YOLO, and normalization was performed to preprocess data.

4.4 Image data augmentation

As a method of image data preprocessing, image augmentation plays an important role in deep learning. In general, effective image augmentation can better improve the robustness of a model and obtain a stronger generalization ability. A large amount of image data is a prerequisite for using deep learning. Normally, before training the model, relevant algorithms are used to increase the size of the dataset. The labor cost of marking supervision training is huge, therefore image augmentation was used in this experiment to reduce labor cost but increase data size.

The samples were upside-down and symmetric-mirror processed, yielding data resources for deep learning implementation. Since the litchi dataset included litchis of different maturity levels, their color characteristics had slight differences. In the process of image augmentation, changing the image color attributes of the dataset would affect the training performance of the network. Therefore, in order to enhance the robustness of the network model without affecting the training performance of the data set, in addition to the operation of changing the image color attributes, random translation, Gaussian noise, flipping, and sharpening operations were used to perform the image augmentation process on the image data set. The image enhancement examples are shown in Figure 10.

a. Original image b. Shift up by 80 pixels c. Shift down by 80 pixels d. Shift right by 80 pixels e. Shift left by 80 pixels

f. Horizontal flip g. Vertical flip h. Noise i. Gaussian blur j. Sharpen

Figure10 Image enhancement examples of litchi

The experiment was evaluated according to the test set, so the validation set was merged with the training set for training. The experiment used partial enhancement rules to randomly extract data from the data set for the corresponding image enhancement. The various enhancement methods and the number of all images after image enhancement are listed in Table 2.

Table 2 The number of training and validation samples treated by different image augmentation methods

Dataset	Original data	Random translational	Gaussian noise	Flip	Sharpen	Total
Training set	2398	500	500	300	100	3798
Validation set	950	500	500	300	100	2350
Total	3348	1000	1000	600	200	6148

4.5 Experimental deployment

This experiment used the DarkNet deep learning framework. The hardware configuration used in this experiment was Intel Core I7-6700 @3.40 GHz X 8 CPU and GeForce GTX TITAN X GPU. The software environment was NVIDIA driver version 390.87, CUDA version 9.0.176, CUDNN version 7.0.5, g++/ GCC version 7.3.0.

This study used batch-wise asynchronous stochastic gradient descent[26] to optimize the processing. Each time 64 images in 8 batches were fed into the network. Each input image is 416 pixels× 416 pixels in RGB color space. The momentum factor was set to 0.9, the attenuation coefficient to 0.0005, and the saturation and exposure to 1.5 times, which easily highlighted the characteristic contrast between the target object and the background. The learning rate was initially set to 0.001, and the maximum number of training iterations was 25 000. The training strategy was to achieve a smaller loss when the learning rate dropped by 0.1 in 19 000 batches and 23 000 batches, respectively. The default anchors were used for prediction, and the jitter coefficient was set to 0.3 to increase the robustness of the model.

4.6 Evaluation

Precision, Recall, F1, and mean Average Precision (mAP) are used to evaluate the recognition performance of the network

designed for litchi. Their calculation method was defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\% \quad (2)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% \quad (3)$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

$$\text{AP}_{\text{C}} = \frac{\sum \text{Precision}_{\text{C}}}{N_C} \quad (5)$$

$$\text{mAP} = \frac{\sum \text{AP}}{N_{\text{Class}}} \quad (6)$$

where, AP_{C} represents the average accuracy of category C. $\text{Precision}_{\text{C}}$ represents the precision of category C on each image; N_C represents the number of pictures containing category C. The numerator portion of Equation (6) represents the sum of AP for all classes, and N_{Class} represents the total number of classes. True Positive (TP), False Negative (FN), False Positive (FP), and True Negative (TN) are used in the above calculation. The differences between them are listed in Table 3. In this experiment, litchi needed to be identified, so litchi fruits were in the positive class, and the others were in the negative class. The F1 score is the harmonic mean of accuracy and recall. The higher F1 is, the better the model performance is.

Table 3 Confusion matrix description

Label	Predicted	Confusion matrix
Positive	Positive	TP
Positive	Negative	FN
Negative	Positive	FP
Negative	Negative	TN

Intersection-over-union (IoU) is the ratio of the intersection and union of "predicted frame (*A*)" and "real frame (*B*)", shown in Equation (7). The larger the intersection area is, the larger the value of IoU is, and the more accurate the detection result is.

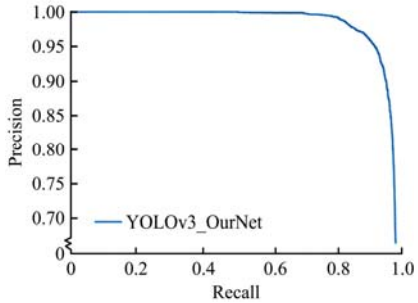
$$\text{IoU}(\text{A}, \text{B}) = \frac{|\text{A} \cap \text{B}|}{|\text{A} \cup \text{B}|} \quad (7)$$

In the subsequent experiments, it is usually necessary to comprehensively weigh indicators such as mAP to illustrate the experimental result. The specific experimental data and experimental methods are summarized in the following sections.

5 Results and discussion

5.1 Experimental result

The experiment set the non-maximum suppression threshold to 0.4 and the IoU threshold to 0.5 to obtain a PR curve, as shown in Figure 11.

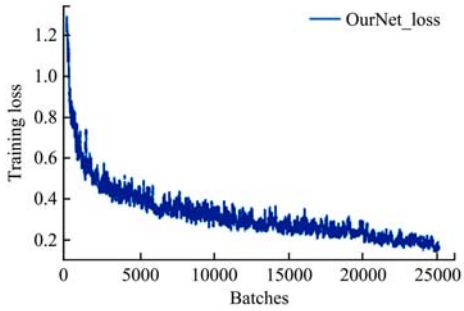


Note: Set the non-maximum suppression threshold to 0.4 and the IoU threshold to 0.5.

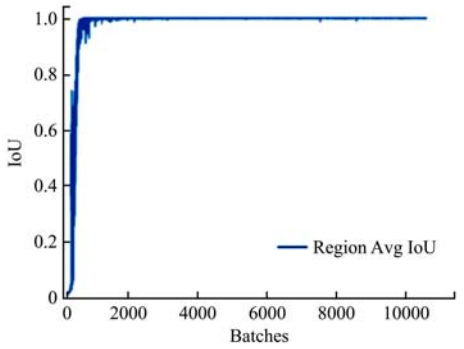
Figure 11 PR curve of YOLOv3_Litchi

As shown in Figure 11, the PR curve tended extremely towards the upper right corner, which demonstrated that the model was ideal for training and that the classifier was better.

Since the loss of the first 1000 iterations was large, this study obtained the loss-batches graph with iterations greater than 1000 for convenience of observation. Figure 12a shows that after 20 000 batches, the loss curve tended to be flat and no longer drops, indicating that the model training was saturated. The graph of the relationship between the IOU and iterations is shown in Figure 12b. The graph shows that the IOU tended towards 1 after training and the accuracy of the model for the training set tended to become saturated.



a. Loss curve



b. IoU curve

Note: OurNet represents YOLOv3_Litchi.

Figure 12 YOLOv3_Litchi's iterations curve

5.2 Comparison of different models

This study compared the performance of the ResNet-50, DarkNet-19 (The backbone network of YOLOV3-Tiny is similar to DarkNet-19), DarkNet-53 (The classic YOLOv3 is based on DarkNet-53, but for the convenience of description, it was called YOLOv3-DarkNet53 in this study) and DesNet-201 networks with the proposed one.

The results of Table 4 show that YOLOv3_Litchi had the highest mAP and the highest detection accuracy. Compared with YOLO_Tiny, the mAP of YOLOv3_Litchi increased by 2.59%. The experimental data shows that the mAP of YOLOv3_Litchi was 1.89% and 1.11% higher than the YOLOv3 model based on DarkNet-53 and DesNet-201 respectively. Moreover, the frame frequency of 58 fps for performance of TITAN X graphics was higher than the results of the classic DarkNet-53 model and DesNet-201 model of the classic YOLOv3. Compared to the classic YOLOv3_DarkNet-53 network, the proposed model in this study had significant improvements in accuracy, model size, and computational cost. Finally, compared with the result of ResNet-50, the mAP increased by 0.86%, and the FPS was higher than YOLOv3_ResNet-50. At the same time, due to the use of lightweight network structure for model training, YOLOv3_Tiny's detection speed could reach 173 fps during detection, but its accuracy was the lowest, unable to meet the needs of litchi

detection in complex scenes. Overall, the comparison of the above experimental data shows that the improvement of this study was meaningful.

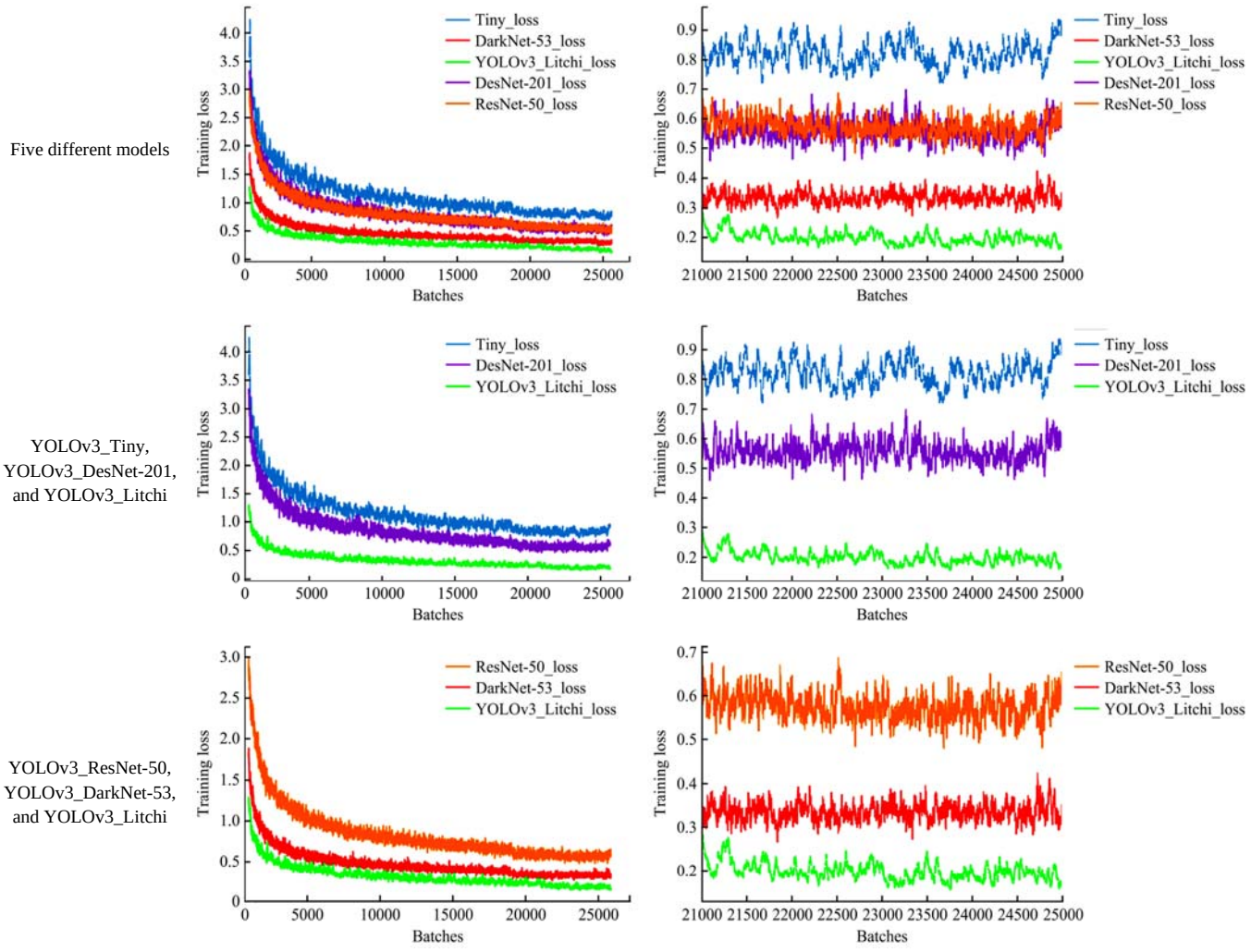
Table 4 Performance of different models comparison

Model	mAP/%	Frame frequency/fps	Size/MB
YOLOv3_ResNet-50	96.21	49	100.7
YOLOv3_Tiny	94.48	**173**	**34.7**
YOLOv3_DarkNet-53	95.18	29	246.3
YOLOv3_DesNet-201	95.96	26	86.9
YOLOv3_Litchi	**97.07**	58	134.7

Note: mAP: mean Average Precision.

By loss comparison among YOLOv3_Litchi, ResNet-50, Tiny, DarkNet-53, and DesNet-201, it can be concluded that YOLOv3_Litchi's losses were significantly smaller than those of other networks, as shown in Figure 13a, and the convergence of the YOLOv3_Litchi was faster. From the enlarged view before the end of training, shown in Figure 13b, the amplitude of YOLOv3_Litchi was smaller than other networks, indicating that the convergence of YOLOv3_Litchi was better than other networks.

YOLOv3_Litchi was compared with two classic models, Faster-RCNN and SSD, as shown in Table 5. Compared with the classic Faster R-CNN using the VGG16 feature-extraction network, the mAP of YOLOv3_Litchi was improved by 1.09%, and the FPS was obviously better. Compared with the classic SSD, the accuracy and FPS were also improved.



a. Loss comparison of 1000-25000 steps b. Loss comparison of 21000-25000 steps

Figure 13 Loss comparisons among different models

Table 5 Performance comparison among different models

Model	mAP/%	Frame frequency/fps	Size/MB
YOLOv3_Litchi	97.07	58	134.7
SSD_VGG16	96.74	50	95
Faster R-CNN_VGG16	95.98	10	483.5

5.3 Different scene comparison experiments

Detection of the occluded target is one of the important factors in verifying the robustness of a detection network. By calculating the proportion of the number of occluded litchis to the total number of litchis in each image, the experiment of this study used 100 images of litchi with sparse distribution and intact individuals without shade, 100 images with less than 50% occlusion, and 100 images with more than 50% occlusion as samples. The comparison results of YOLOv3_DarkNet-53 and the proposed model were shown in Table 6.

Usually, the recall rate and accuracy rate are negatively correlated. The experimental setting of the IoU threshold was 0.5, and the non-maximum suppression value was 0.4, with the experimental results shown in Table 6. It can be seen from Table 6 that the recall rate of the model in this study was better, but its accuracy was relatively low due to the limitation of the non-maximum suppression threshold. The F1 score of the model in this study was higher in the case of less occlusion, and the F1

value decreased as the occlusion increased. However, the improved network showed better performance in terms of accuracy and F1 than that of the YOLOv3_DarkNet-53 network under various occlusion conditions, as shown in Figure 14.

It can be also concluded from Table 6 and Figure 14 that both networks showed a strong performance in sparse and complete fruit image detection, but the improved YOLOv3_Litchi network was more robust and accurate than the YOLOv3_DarkNet-53 network in the case of denser fruit occlusion.

In addition, as shown in Figure 15, the optimized YOLOv3_Litchi network model was robust to litchi detection in complex scenes and had excellent detection performance.

Table 6 Comparison of YOLOv3_ DarkNet-53 and YOLOv3_Litchi under different occlusion conditions

Litchi scenario	Precision		Recall		F1	
	YOLOv3_DarkNet-53	YOLOv3_Litchi	YOLOv3_DarkNet-53	YOLOv3_Litchi	YOLOv3_ DarkNet-53	YOLOv3_Litchi
Sparse and complete	61.97%	67.48%	98.78%	98.92%	0.762	0.802
Occlusion (~, 50)	53.66%	65.66%	98.63%	99.32%	0.695	0.791
Occlusion (50, ~)	43.44%	50.12%	94.54%	97.88%	0.595	0.663
Overall test set	49.57%	59.71%	96.77%	98.23%	0.655	0.743

a. YOLOv3_DarkNet-53 b. YOLOv3_Litchi

Figure 14 Comparison of litchi detection results of this study

a. Dark environment b. Dense occlusion c. Close-up multi-target d. Strong light conditions

e. Rainy scene f. Blacklight g. Green Litchi h. Side-light exposure

Figure 15 Litchi detection results by YOLOv3_Litchi: Litchi with different conditions

6 Conclusions

This study was aimed to detect litchi fruits in field conditions using the combined digital images and deep neural network for precise orchard management. The feature-extraction network was improved by combining dense convolution with the residual network, reducing the number of residual layers. The shallow layer replaced the large convolution with multiple small convolutions. Based on the YOLO regression model, the feature-extraction network of YOLOv3_Litchi was proposed, with an mAP of 97.07% and a detection speed of 58 fps. The experiment results show that the improved network was more robust for occluded targets and achieved better recognition results.

1) The idea of a residual network combined with YOLO regressive detection was proposed. Putting the residual network into the feature-extraction network avoided the problem of precision decrease caused by the excessive depth of the network layer. The mAP of the classic YOLOv3_Darknet-53 was 95.18%, while after the residual structure improvement, the network obtained 97.07% MAP, an increase of 1.89%. Also, the model size was one-half, and the running speed was increased by 29 fps.

2) The combination of dense connection blocks and a residual network was implemented to minimize the loss of shallow-layer features and minimize the model parameters. For the test set with 1400 samples, the F1 value of YOLOv3_Litchi was 0.743, which was higher than the 0.655 of YOLOv3_DarkNet-53.

3) Small convolutions instead of large convolutions for shallow layers of the network ensured the accuracy and receptive field while reducing the model parameters. The nonlinearity was improved by the 1×1 convolution kernel while ensuring the constant size of the feature map. Partial improvements to the ResNet network had reduced the size of the model. Compared with the YOLOv3 model based on ResNet50, the proposed model had an improved detection speed by 9 fps and an improved mAP.

4) Based on the characteristics of the litchi data set, the improved FPN structure was used to combine the high-resolution feature map with the low-resolution high-semantic feature, which had a better recognition and positioning effect for small targets.

In YOLOv3_Litchi, the fusion of up-sampling and shallow features was carried out, and the shortcoming of poor detection performance for small targets was solved, resulting in improvement of the performance of the detection network. With agricultural development, litchi automatic picking is an inevitable trend in the future for the litchi production industry. An accurate, stable, real-time efficient fruit localization algorithm is key in the fruit positioning technology of picking robots. In this study, the feature-extraction network was improved by the proposed method, obtaining a higher and faster target detection. The findings provided powerful technical support for the target detection and location by the picking robots.

Acknowledgments

This work was financially supported by the National Natural Science Foundation of China (Grant No. 32071912; No. 61863011; No. 31701325; No. 31571568; No. 31570180), the Guangzhou Science and Technology Project (Grant No. 202002020016; No. 202102080337), the Natural Science Foundation of Guangdong Province (Grant No. 2018A030313330; No. 2020A1515010793), the Second Batch of Industry-Education Cooperation Collaborative Projects in 2019, Ministry of Education (Grant No. 201902062040), the Guangzhou Key Laboratory of Intelligent Agriculture (Grant No. 201902010081), the Project of Rural Revitalization Strategy in Guangdong Province (Grant No. 2020KJ261), and the Applied Science and Technology Special Fund Project, Meizhou, China (Grant No. 2019B0201005).

[References]

[1] Wei X Q, Ji K, Lan J H, Li Y W, Zeng Y L, Wang C M. Automatic method of fruit object extraction under complex agricultural background for vision system of fruit picking robot. Optik, 2014; 125(19): 5684–5689.

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

[3] Tao Y T, Zhou J. Automatic apple recognition based on the fusion of color and 3D feature for robotic fruit picking. Computers and Electronics in Agriculture, 2017; 142(Part A): 388–396.

[4] Ni X D, Wang X, Wang S M, Wang S B, Yao Z, Ma Y B. Structure design and image recognition research of a picking device on the apple picking robot. IFAC-PapersOnLine, 2018, 51(17): 489–494.

[5] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. Communications of the ACM, 2012; 60: 84–90.

[6] Szegedy C, Liu W, Jia Y Q, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, USA: IEEE, 2015; pp.1–9. doi: 10.1109/CVPR.2015.7298594.

[7] Peng H X, Xue C, Shao Y Y, Chen K Y, Xiong J T, Xie Z H, et al. Semantic segmentation of litchi branches using DeepLabV3+ model. IEEE Access, 2020; 8: 164546–164555.

[8] Kang H W, Zhou H Y, Wang X, Chen C. Real-time fruit recognition and grasping estimation for robotic apple harvesting. Sensors, 2020; 20(19): 5670. doi: 10.3390/s20195670.

[9] He, K., Zhang, X., Ren, S. and Sun, J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas: IEEE, 2016; pp.770–778.

[10] Sa I, Ge Z Y, Dayoub F, Upcroft B, Perez T, McCool C. DeepFruits: A fruit detection system using deep neural networks. Sensors, 2016; 16(8): 1222. doi: 10.3390/s16081222.

[11] Bargoti S, Underwood J P. Image segmentation for fruit detection and yield estimation in apple orchards. Journal of Field Robotics, 2017; 34(6): 1039–1060.

[12] Zhang Y D, Dong Z C, Chen X Q, Jia W J, Du S D, Muhammad K, et al. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. Multimedia Tools and Applications, 2019; 78: 3613–3632.

[13] Kestur R, Meduri A, Narasipura O. MangoNet: A deep semantic segmentation architecture for a method to detect and count mangoes in an open orchard. Engineering Applications of Artificial Intelligence, 2019; 77: 59–69.

[14] Ren S Q, He K M, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016; 39(6): 1137–1149.

[15] Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016; pp.779–788.

[16] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C Y, et al. SSD: Single Shot MultiBox Detector. In: European Conference on Computer Vision-ECCV 2016, Springer, 2016; 9905: 21–37. doi: 10.1007/978-3-319-46448-0_2.

[17] Tian Y N, Yang G D, Wang Z, Wang H, Li E, Liang, Z Z. Apple detection during different growth stages in orchards using the improved YOLO-v3 model. Computers and Electronics in Agriculture, 2019; 157: 417–426.

[18] Zhao J Y, Qu J H. Healthy and diseased tomatoes detection based on YOLOv2. In: International Conference on Human Centered Computing-HCC 2018, Springer, 2018; 11354: 347–353. doi: 10.1007/978-3-030-15127-0_34.

[19] Liu G X, Nouaze J C, Touko Mbouembe P L, Kim J H. YOLO-Tomato: A robust algorithm for tomato detection Based on YOLOv3. Sensors, 2020(7); 2145. doi: 10.3390/s20072145.

[20] Neubeck A, Gool L J V. Efficient non-maximum suppression. In: 18th International Conference on Pattern Recognition (ICPR'06), HongKong, China: IEEE, 2006; pp.850–855. doi: 10.1109/ICPR.2006.479.

[21] He K M, Zhang X Y, Ren S Q, Jian S. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016; pp.770–778. doi: 10.1109/CVPR.2016.90.

[22] Huang G, Liu Z, Van Der Maaten L, Weinberger K Q. Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, USA: IEEE, 2017; pp.2261–2269. doi: 10.1109/CVPR.2017.243.

[23] in T Y, Dollár P, Girshick R, He K, Hariharan B. Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, USA: IEEE, 2017; pp.936–944. doi: 10.1109/CVPR.2017.106.

[24] Ye M, Liu G Y. Facial expression recognition method based on shallow small convolution kernel capsule network. Journal of Circuits, Systems and Computers, 2020; 30(10): 2150177. doi: 10.1142/S0218126621501772.

[25] Tek F B, Çam I, Karlı D. Adaptive convolution kernel for artificial neural networks. Journal of Visual Communication and Image Representation, 2021; 75: 103015. arXiv: 2009.06385.

[26] Goyal P, Dollár P, Girshick R, Noordhuis P, Wesolowski L, Kyrola A, et al. Accurate, large minibatch SGD: Training ImageNet in 1 hour. 2017. arXiv: 1706.02677v1.