High-Quality Coarse-to-Fine Fruit Detector for Harvesting Robot in Open Environment

Li Zhang[1,2], YanZhao Ren[1,2], Sha Tao[1,2], Jingdun Jia[1,3*], and Wanlin Gao[1,2*]
[1] College of Information and Electrical Engineering, China Agricultural University
Beijing, 100083, China
[e-mail: zhanglsky@126.com]
[2] Key Laboratory of Agricultural Informatization Standardization, Ministry of Agriculture
Beijing, 100083, China
[e-mail: xiaozhaochina@163.com, taosha20070608@163.com, wanlin_cau@163.com]
[3] National Rural Technology Development Center, Ministry of Science and Technology
Beijing, 100862, China
[e-mail: jiajd@most.cn]
*Corresponding author: Wanlin Gao, Jingdun Jia

Received December 7, 2020; revised January 24, 2021; accepted February 5, 2021; published February 28, 2021

Abstract

Fruit detection in orchards is one of the most crucial tasks for designing the visual system of an automated harvesting robot. It is the first and foremost tool employed for tasks such as sorting, grading, harvesting, disease control, and yield estimation, etc. Efficient visual systems are crucial for designing an automated robot. However, conventional fruit detection methods always a trade-off with accuracy, real-time response, and extensibility. Therefore, an improved method is proposed based on coarse-to-fine multitask cascaded convolutional networks (MTCNN) with three aspects to enable the practical application. First, the architecture of Fruit-MTCNN was improved to increase its power to discriminate between objects and their backgrounds. Then, with a few manual labels and operations, synthetic images and labels were generated to increase the diversity and the number of image samples. Further, through the online hard example mining (OHEM) strategy during training, the detector retrained hard examples. Finally, the improved detector was tested for its performance that proved superior in predicted accuracy and retaining good performances on portability with the low time cost. Based on performance, it was concluded that the detector could be applied practically in the actual orchard environment.

Keywords: Fruit Detection, Coarse-to-Fine, Synthetic Dataset, Harvesting Robot, Multi-task Cascaded

1. Introduction

Due to the rapid development of autonomous robots and precise agriculture, more and more automation equipment is becoming available in the field of agriculture [1, 2]. Generally, there are two reasons for this rapid development. One is that the use of automated robots in agriculture greatly reduce labor cost, and bring more profit for the enterprise [3-7]. The other reason is the strong demand for unified quality and standard agricultural products. Therefore, studies initiated on automated robots to facilitate agricultural production decades ago [8-10]. In this regard, high accuracy and low time costing fruit detection methods are required for a variety of follow-up works.

Fruit detection in uncertain and unrestrained orchards indubitably encounters numerous challenging tasks, such as variation of the pose, low-resolution, heavy occlusion by neighboring fruits or foliage, indistinguishable backgrounds, insufficient or over-illumination, and so on. In recent years, studies have been conducted to develop robust fruit detection algorithms based on the high performance of deep learning. On one hand, although many deep learning-based methods outperform very well compared with traditional ones, a sufficient and diverse dataset is an inevitable important for most of these methods. More seriously, it is hard to use one dataset as a standard benchmark, due to the different varieties and growth environment of fruits in the wild of orchards. Flexible and simple methods to set up a dataset is required for the promotion to practical application. On the other hand, many studies regard fruit detection as conventional object detection, so most of these studies are based on classical object detection architecture which achieves remarkable results. However, in some cases, these methods have a deficiency in considering the number of parameters and the prediction time cost. Due to the methods with very deep layers and a large number of parameters is pessimistic to be deployed to automatic robots. The main contributions of the paper are summarized as follows:

- This is the first attempt to generate a synthetic dataset for fruit detection. Only a few manual operations are needed to generate a large number of random diversities that are very close to the samples captured in a real scene environment. The detection model trained with a supplemental synthetic dataset greatly improves the results.
- We improved fruit detection based coarse-to-fine multi-task cascaded convolutional network (Fruit-MTCNN) [11] architecture by applying center loss function. This was the first attempt of using the center loss to increase the inter-class variations in fruit detection. This novel architecture is named as Fruit-MTCNN-Imp, for short. Moreover, the OHEM strategy was used during the model training to level up the discrimination power of the fruit detector by strengthening the learning of prone to false predicted images.
- Extensive quantitative and qualitative evaluations demonstrate the proposed synthetic dataset and improved detection methods are effective for fruit detection task in the wild of orchards.

The rest of the paper is organized as follows. Section 2 introduces the background and related work. Section 3 describes both conventional and synthetic datasets used in this study, and the method how to generate synthetic images is described in detail. Section 4 of this paper presents the improved architecture and training strategy of Fruit-MTCNN-Imp. The experimental results are demonstrated in section 5. Section 6 encompasses a detailed discussion of the experimental results. Conclusions and the future work plan are shown in section 7.

2. Related Work

Fruit detection is one of the most essential tasks in an automated robot visual system. It has been studied for decades and has gained remarkable progress. In this section, the background and some previous efforts made in this area are enumerated.

2.1 Traditional Methods

Traditional methods for fruit detection can be divided into two categories, one is based on image processing technology, as mentioned by [12-16]. To detect crops from images, [12] based on the assumption that the color of the crop maintained within one or several simply connected domains, [13] proposed an algorithm based on image processing technology to convert RGB color space to CIE-Lab values for inspecting the surface color of two Thai mango cultivars, [15] detected fried potato chips by extracting discriminatory features in the continuous wavelet transform domain using Morlet wavelet. [16] also developed an automatic tomato grading system based on image processing technology. In general, all the above methods are designed only for a specific task, highly dependent on the characteristics of the subjects which need to be re-designed when there is only a slight change in conditions. Therefore, the disadvantages of such kinds of methods are highly dependent on a certain condition, prone to reduce accuracy drastically when met tiny changes.

The other category is based on machine learning technology reported by [17-23]. Based on the consideration of choosing the proper time, [17] explored a method based on k-Means clustering to choose a proper time. [20] used naive Byes and support vector machine (SVM) for grading mangoes into three categories. The advantages of these methods are exhibit higher accuracy and more robust stability compared with the methods based on image processing, when face slight changes. However, these kinds of methods are needed to extract proper features by experienced experts, such as surface pixel, color, shape, and size. Therefore, there is still a long distance to get them promoted to the level of practical use.

2.2 Deep Learning based Methods

During the recent years, deep learning based methods have made remarkable progress in many fileds [24], such as Internet of Things [25, 26], Signal processing [27, 28], UAV [29], wireless communications [30], and especially in the field of agriculture [31-35]. These include fruit classification [36-38], yield estimation and counting [39, 40]. Overall, the prevailing deep learning based fruit detection methods can be divided into two categories, one is based on the two-stage structure of faster region-based convolutional neural networks (Faster-RCNN) [41], such as [42-49]. [42] presented an algorithm to detect and classify passion fruits based on maturity. They trained Faster-RCNN on RGB data and depth data, and fused these two detectors to make an RGB-D detector to achieve higher accuracy for passion fruits detection. [43] presented a novel method using color and thermal images to Faster-RCNN architecture to detect immature green citrus fruits. [44] presented a Faster-RCNN based approach for fruit detection by exploiting both color images and Near-Infrared images, and then explored the two information methods for early and late fusion. Similarly, [45] also used Faster-RCNN for mango, almond and apple detection in orchards. [45-49] also present a benchmark dataset for apple detection, which largely reduce the task of image collection for deep CNNs training. The above methods based on Faster-RCNN achieved leap forward improvements. However, the higher time cost is one of weakness, which may limit these methods promoted to the practical application.

Another one is based on one-stage structure, such as the series of you only look once [50, 51] (YOLO). [52] exploit YOLO for real-time apple detection with tree. [53] explored YOLO as a baseline model to detect apples during different growth stages. Such kind of one stage fruit detector with less time cost compared with methods based on the two stages, although slightly inferior in the aspect of accuracy. It may due to that both the architecture of Faster-RCNN and YOLO series are proposed for the multi-class objects detection, so they have deep network layers and a large number of parameters which beyond the hardware capacity of automatic robots. Therefore, a more suitable fruit detection system is needed to be developed under such circumstances.

3. Description of Dataset

Although [45-49] presented benchmark for apple detection, similarly environment factor (e.g., resolution of camera, varieties, illumination, the distance, angle between camera and the target objects etcetera.) for each benchmark setup will weak the generalization ability of the trained model to a new environment. Due to these reasons, this paper generates a synthesis dataset which only need few samples.

3.1 Image Acquisition and Conventional Dataset

A Canon EOS 100D digital camera was used to capture nearly 1800 images from an apple orchard. During image acquisition, we changed the camera's viewing angles and shooting distance to collect diverse samples. Besides, our method applied training data from Internet and ImageNet (an open-source database) [54] which were easily acquired. All the objects were labeled manually as individual image datasets. According to the density of objects, divided these images into training and test datasets randomly with a ratio nearly three to one.

3.2 Preparation of Elements for the Synthetic Dataset

Considering the ever-changing orchard environment, more diversified samples are beneficial for the detection network training and final result. On the contrary, collection and annotation of images to set up a sufficient dataset is a time-consuming and boring task. Therefore, an attempt was made to find methods to simplify the process in order to generate image data close to the images captured in the real environment. In this study, a novel method is described to generate dataset only with a few labeled images that would further create images highly similar to those obtained in the real environment [55]. The overall flowchart is shown in **Fig. 1**.

3.3 Background and Objects Description

In this study, scene images of an apple orchard were randomly collected as background images. These images included orchard environment such as sky, ground, apple trees, etc. For foreground scene, objects were widely divided into two categories i.e., positive objects and negative objects. Positive objects indicate the objects wanted to be detected, so diverse viewpoints of apple images were chosen as positive object images. Negative objects indicate the objects with high probability to influence positive objects. Image data were generated as close to the real data as possible, so multi-view point branch and leaf object were collected as negative objects. The number of each category are shown in **Table 1**.

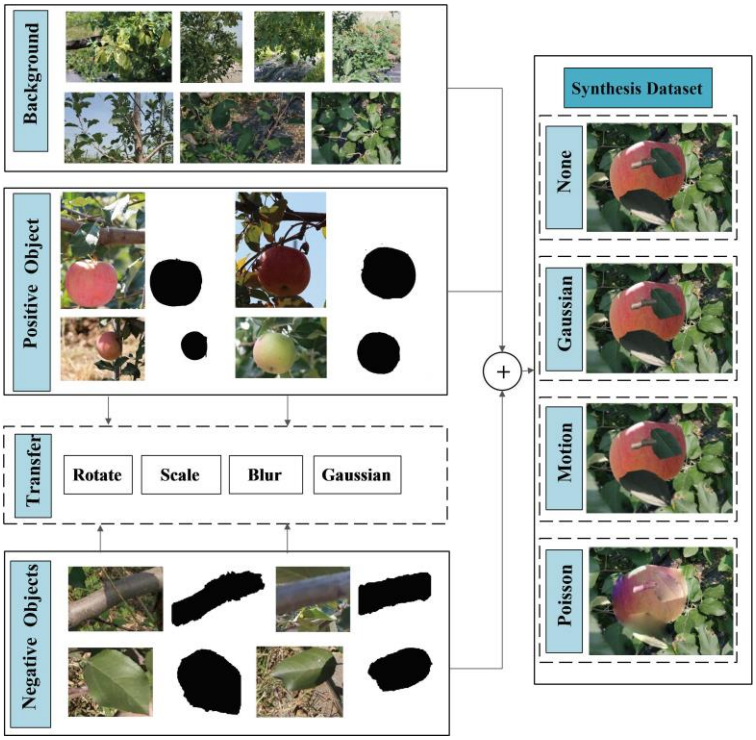

Fig. 1. The flowchart of generating synthesis dataset

In **Fig. 1**, is four parts are needed to generate synthesis images. They are background images, positive object images, transfer operation, and negative object images. Particularly, each object with one RGB image and one corresponding black-white mask image. Transfer operation includes rotating transformation, scale transformation, Gaussian noise, and a blur of motion and Poisson. Four synthesis images are generated each time, the original image (None), the image with Gaussian noise (Gaussian), the image with Motion blur (Motion) and the image with Poisson blur (Poisson).

Table 1. The Object Number of Each Category

Name of each category			Number
Foreground	Positive Object	Fruits	132
		Leaves	35
	Negative Object	Branches	12
Background Image			29

3.4 Object Segmentation

There are many deep learning-based segmentations network models that have achieved outstanding performance. Generally, these methods suit to a variety of different categories. In this study, only four categories needed to be distinguished i.e., background, apple fruits, leaves,

and branches. Based on these considerations, the GrabCut method was adopted for image segmentation.

GrabCut is an image segmentation method derived from the GraphCut algorithm [56]. In this case, based on the specified bounding box of the object to be segmented, a Gaussian mixture model (GMM) is used to estimate the color distribution of the object and its background. This algorithm can achieve the optimal segmentation of foreground and background with a few labeled pixels. For comparing contrast texture or color information in the image, this algorithm can achieve a better segmentation effect with only a few user interaction operations and with high accuracy performance. Based on the above considerations, GrabCut algorithm was deemed the most appropriate choice for this work.

Here an energy function E is defined so that its minimum should correspond to a good segmentation. It is guided by both the observed foreground and background grey-level histograms and that the opacity is "coherent", reflecting a tendency to the solidity of objects. As shown in (1):

$$E(\underline{\alpha}, k, \underline{\theta}, z) = E(\underline{\alpha}, k, \underline{\theta}, z) + V(\underline{\alpha}, z) \tag{1}$$

where the data term U evaluates the fit of the opacity distribution to the data z, given the histogram model q, and is defined as in (2):

$$U(\underline{\alpha}, k, \underline{\theta}, z) = \sum D(\alpha_n, k_n, \underline{\theta}, z_n) \tag{2}$$

Each GMM, i.e., one for the background and one for the foreground, is taken to be a full-covariance Gaussian mixture with K components (typically K = 5). $K = \{k_1, \ldots, k_n, \ldots, k_N\}$ and $k_n \in \{1, \ldots, N\}$ assigning, to each pixel, a unique GMM component, one component either from the background or the foreground model, according to α_n =0 or α_n =1. D is defined in (3) as:

$$\begin{aligned} D(\alpha_n, k_n, \underline{\theta}, z_n) &= -\log \pi(\alpha_n, k_n) + \frac{1}{2} \log \det \sum(\alpha_n, k_n) + \\ &\frac{1}{2}\left[z_n - \mu(\alpha_n, k_n)\right]^T \sum(\alpha_n, k_n)^{-1} \left[z_n - \mu(\alpha_n, k_n)\right] \end{aligned} \tag{3}$$

and $\underline{\theta} = \{\pi(\alpha, k), \mu(\alpha, k), \sum(\alpha, k), \alpha = 0,1, k = 1 \ldots K\}$ where the weights π, means μ, and covariance $\sum$of the 2K Gaussian components for the background and foreground distributions. Finally, the smoothness term V is defined in (4) as:

$$\mathrm{V}(\underline{\alpha}, z) = \gamma \sum_{(m,n) \in C} \left[a_n \neq a_m\right] \exp - \beta \left\| z_m - z_n \right\|^2 \tag{4}$$

We used the GrabCut method to get some object for 'paste'. Each positive or negative objects should have a pair of images; one is this segmented object color image and the other is black-white binary mask image. We only needed to label a few pixels to generate corresponding images and some segmented objects as shown in **Fig. 2**.

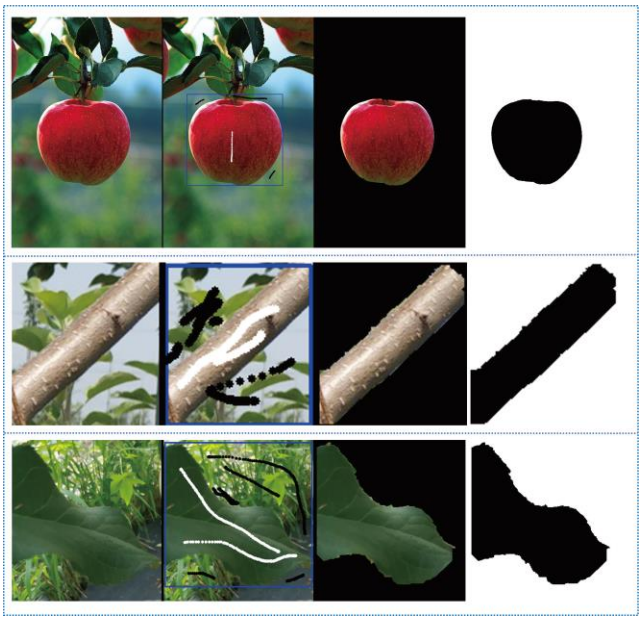

Fig. 2. Few steps to generate segmented objects

In **Fig. 2**, the first column (starting from left) is the original images of the objects. The second column is recorded after manual annotation, which includes two steps, the first step is to segment the object from the background with a blue rectangle, then label background (black line) and foreground (white line). The third column is the result after the GrabCut segmentation algorithm. The final column represents the corresponding masked images of the object.

3.5 Synthesis Dataset

After being prepared with each necessary element, we generated synthesis image data by 'pasting' the images of the positive or negative object to the background image. The corresponding annotation file generated at the same time. **Table 2** presents the algorithm to generate synthesis dataset, such synthetic images were applied in the later training and evaluation which proved to improve final outputs effectively. According to occlusion conditions, specifically four categories of synthesis images were generated, i.e., 1. without occlusion and negative object, 2. without occlusion and with negative object, 3. with occlusion and without negative object, and 4. with occlusion and with negative object. **Fig. 3** shows some examples of synthesis image data.

Further, synthetic image generation was extended through augmentation methods. For generating one original synthetic image, three types of augmented images i.e., Gaussian blur, Motion blur, and Poisson were blended. A total of 3110 synthetic data were generated. The final data generated for each category are shown in **Table 3**. In addition, it's worth note that these synthesized images only used as a complementary part of the training dataset.

Table 2. Algorithm to generate synthesis dataset

Require: Initialization:
Ensure:
1. The foreground set $T_F = \varnothing$, background $T_U = \overline{T_B}$, T_B is initialized by the user.
2. Initialize $\alpha_n = 0$ for $n \in T_B$ and $\alpha_n = 1$ for $n \in T_U$

3. $\alpha_n = 0$ and $\alpha_1 = 1$ for background and foreground GMMs initializing respectively.

Require:User editing

Fix some pixels $\alpha_n = 0$ as background, fix some pixels $\alpha_n = 1$ as foreground brush; update map (T) accordingly (this step can be performed during the entire iterative minimization algorithm) .

Iterative minimization

1. For $i = 1\ to\ n$, do:

2. Assign GMM components to pixels: for each n in T_U

$$k_n := \arg\min_{k_n} D_n\left(a_n, k_n, \theta, z_n\right).$$

3. Learn GMM parameters from data z :

$$\underline{\theta} := \arg\min_{\underline{\theta}} U\left(\underline{a}, k, \underline{\theta}, z\right).$$

4. Estimate segmentation: use min cut to solve:

$$\min_{\{a_n : n \in T_U\}} \min_{K} E\left(\underline{\alpha}, k, \underline{\theta}, z\right).$$

5. End for

6. Apply border matting.

7. Save the corresponding mask file.

Fig. 3. Samples from synthetic dataset

In **Fig. 3**, the first, second, third, and fourth rows (from top to bottom) represent images with one, two, three, and four positive objects (fruits) instances respectively. The first and the second columns (from left) present some examples generated without occlusion and negative samples. The third and fourth columns represent the images generated with occlusion and negative sample influence.

Table 3. The number of final data generated for each category

Scale	Rotation	Occlusion	Negative samples	number
YES	YES	NO	NO	695
YES	YES	YES	NO	765
YES	YES	NO	YES	715
YES	YES	YES	YES	935

4. Our Proposed Improved Fruit-MTCNN Method

Although Fruit-MTCNN detector provided good performance, it still needed improvement to reduce the negative result overall. Therefore, a deeper investigation of the Fruit-MTCNN framework was conducted with two aspects. One aspect was based on a newly developed multi-task loss function in the Fruit-MTCNN model to deeply discriminate features for fruit or non-fruit classification. The other aspect was based on the online hard example mining method during training to generate hard samples for intensive learning. The detailed description of the improved Fruit-MTCNN is elaborated as follows.

4.1 Fruit-MTCNN based Architecture

The Fruit-MTCNN architecture was used as the baseline framework, as shown in **Fig. 4**. Fruit-MTCNN consisted of three sub-networks. They are proposal network (PNet), refine network (RNet), and output network (ONet) respectively.

In the first stage, a large number of coarse detectors were generated from PNet and then passed to RNet as inputs. In the second stage, RNet refined coarse candidate windows and bounding boxes from PNet. Finally, such refined candidate windows from RNet passed to ONet for further step refinement processing. With these three cascaded tiny networks, the fruit detectors getting from coarse to fine values with very low computation requirement which could be applied to harvesting robots easily.

4.2 Loss Function

Motivated by the familiar analysis features of false positive and false negative samples predicted by Fruit-MTCNN deeply, we considered strengthening the ability of this detector to distinguish between the class of fruit and its background. Considering that PNet played a primary role in providing the initial candidate bounding box, we only improved the loss function of RNet and ONet. They are expressed as follows in (5):

$$Loss(p,t,x) = \alpha L_{cls}(p, p^*) + \beta L_c(x) + \gamma L_{reg}(t, t^*) \tag{5}$$

where $L_{cls}(p,p^*)$, $L_c(x)$, $L_{reg}(t,t^*)$ represent softmax loss, center loss, and regression loss respectively. In this paper, the values of the three coefficients α, β, γ are 0.45, 0.05, and 0.5 respectively. p, p^* present the ground-truth value and the estimated value of the probability respectively. t, t^* present the ground-truth value and the estimated value of the bounding box value respectively. x shows the points for a center loss. The loss function of PNet was kept the same with Fruit-MTCNN, expressed as (6):

$$Loss(p,t,x) = \alpha L_{cls}(p, p^*) + \gamma L_{reg}(t, t^*) \tag{6}$$

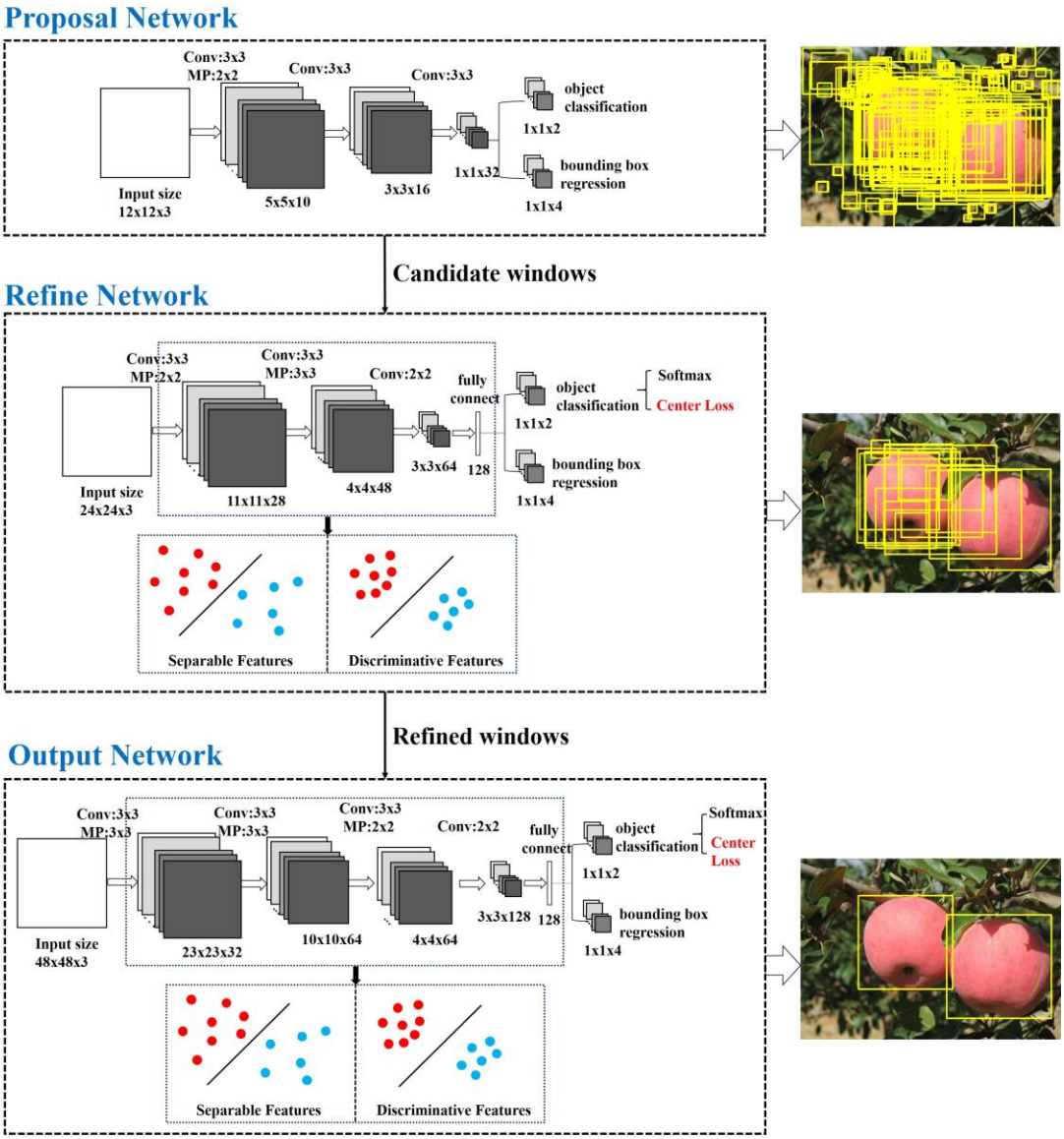

Fig. 4. The overall architecture of the coarse-to-fine fruit detector. From top to bottom, they are PNet, RNet and ONet respectively. "Conv" is convolution and "MP" represents max pooling.

4 where $L_{cls}(p,p^*)$ and $L_{reg}(t,t^*)$ are softmax loss 206 and regression loss respectively α =0.5 and γ =1.0.

4.2.1 Center Loss

By learning a center for deep features of each class and penalizing the distances between the deep features and their corresponding class centers, center loss with good performance in distinguishing between each class [57] was acquired as per (7) as follows:

$$\frac{\partial L_c}{\partial x_i} = x_i - c_{yi}, \Delta c_j = \frac{\sum_{i=1}^{m} \delta(y_i = j) * (c_j - x_i)}{1 + \sum_{i=1}^{m} \delta(y_i = j)} \tag{7}$$

due to the difficulties of updating the entire training set, this operation taken on a min-batch m. L_c , c_{yi} are presented as the gradients and update equation respectively. Where $c_{y_i} \in R^d$ indicates the y_i th class center of deep features, the $x_i \in R^d$ indicates the deep features belonging to the y_i th class. If y_i equals to j, $\delta(y_i = j) = 1$, otherwise $\delta(y_i = j) = 0$. So that the value will be restricted to [0, 1] to control the learning rate of the centers.

4.2.2 Softmax Loss

The classification task is to distinguish fruits from the background, so it can be regarded as a two-class classification problem. Cross-entropy loss is exploited for each sample x_i. Thus, the softmax loss function is shown in (8):

$$L_i^{cls} = -(y_i^{cls} \log(p_i) + (1 - y_i^{cls})(1 - \log(p_i))) \tag{8}$$

where $y_i^{cls} \in \{0,1\}$ presents ground-truth value, p_i is the probability of the input sample x_i, being a fruit.

4.2.3 Regression Loss

The regression loss is applied to align the fruit detectors to the ground-truth values during the training. There are four coordinates for each bounding box, and the regression loss function is shown in (9).

$$L_i^{reg} = \left\| \hat{y}_i^{reg} - y_i^{reg} \right\|_2^2 \tag{9}$$

where $\hat{y}_i^{reg}$ denotes the predicted values by the detector and $\hat{y}_i^{reg}$ is the corresponding ground-truth value.

4.3 Online Hard Example Mining

One way of strengthening the discrimination ability of the fruit-detector is by re-learning the hard examples, which failed to predict. In this study, the OHEM method was employed that included three steps [58]. First, all the true negative samples were collected and false-positive samples were predicted by the detector. Then, in order to balance the positive samples and negative samples, a ratio of 1:1 was set in each mini-batch. Finally, after each SGD iteration, the selected hard examples were fed to the network in the next iteration by a forward pass through the current network.

5. Experiments

5.1 Evaluation Metrics

In the present study, precision rate (P) and recall rate (R) were utilized as evaluation methods for fruit detection. The P and R were computed as per the formulae shown in (10):

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN} \tag{10}$$

where *TP* presents the number of the correct detection results. *FP* is the false detection number, and *FN* is the number of missing objects. The F1 score was also used to evaluate the performance of the model. The definition of the F1 score is shown as follows in equation (11):

$$F1 = \frac{2 \times P \times R}{P + R} \tag{11}$$

5.2 Influence of Different Architectures

The proposed improved detector (Fruit-MTCNN-Imp) was compared with a previous detector (Fruit-MTCNN) to verify the performance of its improved architecture. All the other conditions that could possibly influence were kept the same. These influence factors include using the same dataset for training and testing, the number of iterations for each subnet (PNet, RNet, and ONet) during training, and the same threshold value for testing.

The P-R curves for these two detectors are shown in **Fig. 5**, and some test results are given in **Fig. 6** Based on the experimental results, it was observed that the proposed Fruit-MTCNN-Imp showed significant improvements for the final detection task over the previous version.

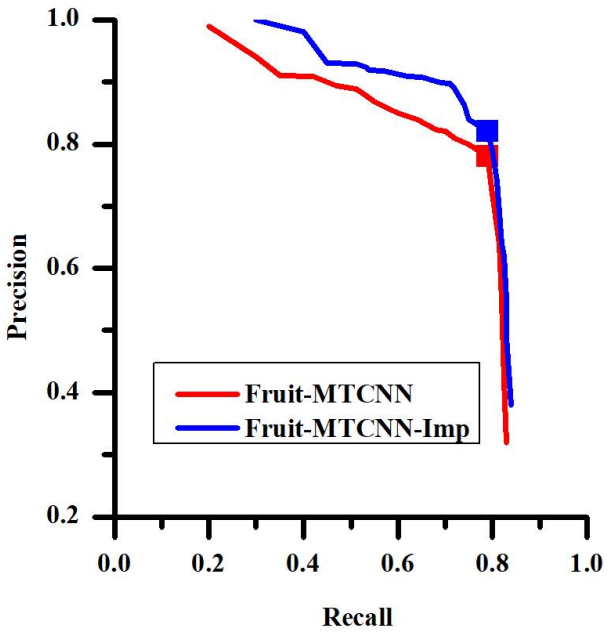

Fig. 5. P-R curves predicted by detection models with different architecture and the square points are presented as each F1-score.

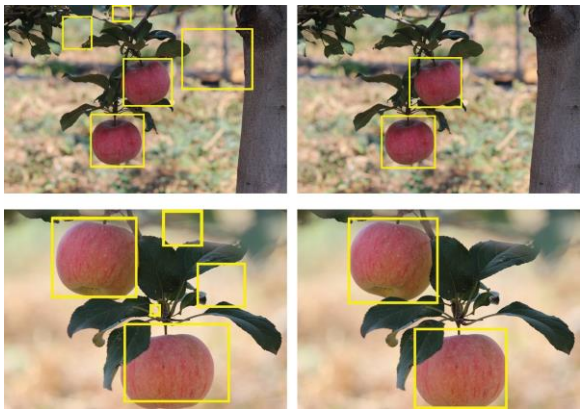

Fig. 6. Some examples predicted by the two detectors. The left column shows results predicted by Fruit-MTCNN. The right column depicts the results predicted by the Fruit-MTCNN-Imp detector for the same images.

5.3 Influence of Synthetic Dataset

In order to verify the effect of the synthetic images during the training of the model, three datasets i.e. conventional, synthetic, and conventional ones mixture with synthetic images (O-S) were created. Firstly, conventional and synthetic datasets were achieved through the methods presented in section 3 and section 4 respectively. After that, the O-S dataset was created by mixing all of the images from the conventional and synthetic datasets. Moreover, we randomly divided each dataset into a training set and a test set by the ratio of 6 to 4.

Finally, the Fruit-MTCNN-Imp model was trained on these three datasets. During training, the number of iterations for each subnet (PNet, RNet, and ONet) was kept the same. The threshold value was also kept the same during the test. During Vthe test of each model, P-R curves were also drawn as shown in **Fig. 7**. **Fig. 8** depicts some typical samples predicted by these detectors.

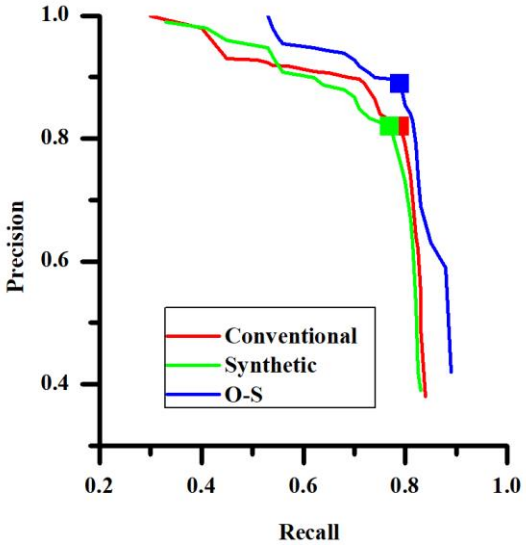

Fig. 7. P-R curves predicted by different detection models trained on different datasets and the square points are presented as F1-score.

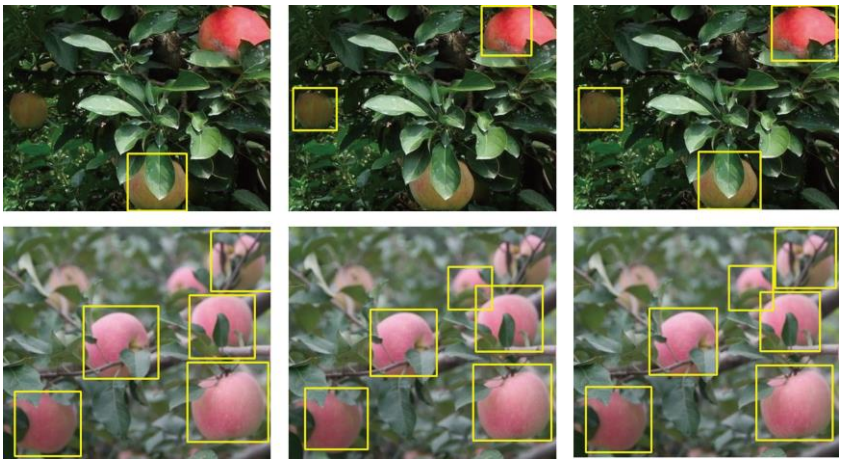

Fig. 8. Some examples predicted by improved Fruit-MTCNN trained on different datasets.

In **Fig. 8**, the left column is the results of the detector trained on the original dataset. The middle column is the results of the detector trained on synthesis dataset. The right column is the results of the detector trained on the mixed dataset (original dataset mixed with synthesis dataset).

The P-R curves and the F1 score values show that the model trained on the O-S dataset was significantly superior to that trained on the other two datasets. The performance of the model trained on the conventional dataset was close to the model trained on a synthetic dataset, which indicated that the synthesized image data set was very close to the images captured in a real environment.

5.4 Influence of OHEM

In order to verify the influence of OHEM strategy on the performance and final prediction of the model, one Fruit-MTCNN-Imp architecture was built with OHEM strategy and the other one without it. The rest of the factors were kept the same whilst training these two models. Finally, P-R curves predicted during the test of each model were drawn, as shown in **Fig. 9**.

From the P-R curves, it is obvious that the model trained with the OHEM strategy substantially improved the final detection result.

5.5 Influence of different levels of overlaps

In this section, the impact of the severity of occlusion is analyzed. The test dataset was divided into three levels according to the severity of occlusion. These were light, medium, and heavy occlusions. P-R curves predicted by the pre-trained model for these datasets were drawn, as shown in **Fig. 10**.

The results indicate that for the objects with lighter occlusion, the detector predicted with higher accuracy. On the contrary, with heavier occlusion, the detection accuracy was severely affected.

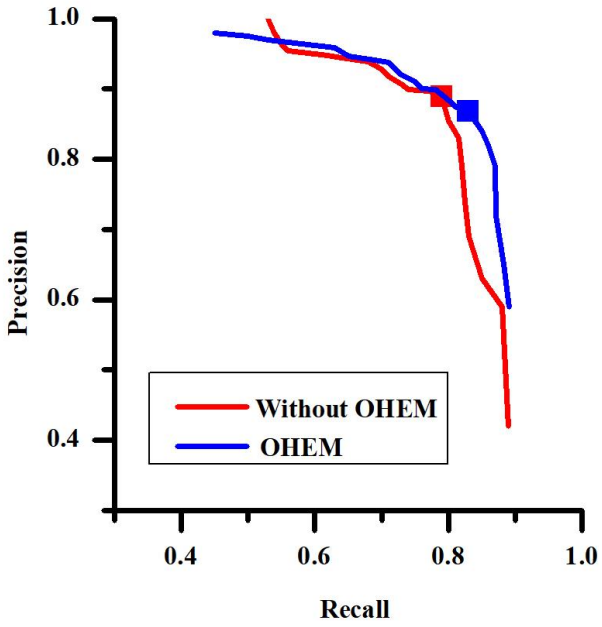

Fig. 9. P-R curves predicted by detection models trained with or without OHEM strategy, and the square points are presented as F1-score.

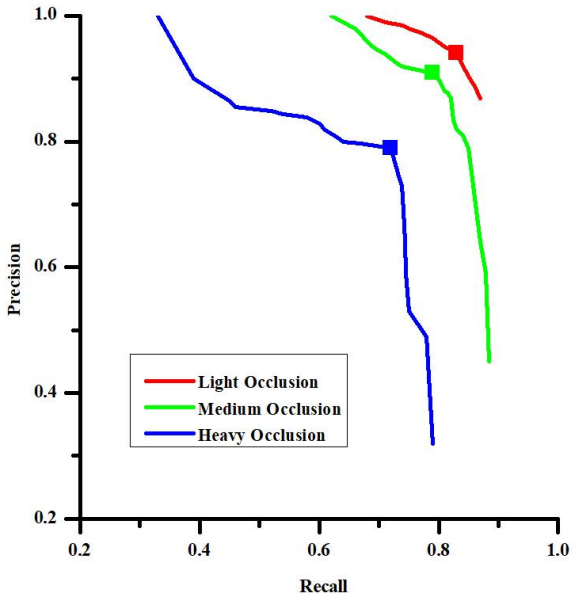

Fig. 10. P-R curves of apple detection models for different severity levels of occlusion, and the square points are presented as F1-score.

6. Discussion

Previous work in the field of fruit detection has some substantial achievements in terms of paving the way towards designing automated robots. However, there was still space for further improvement before these models could be successfully used in the field of agriculture. It was clear scope for improvement in the architecture of the model in order to make it more suitable for use in horticultural production. Therefore, this study attempted to improve the performance of the model by manipulating some vital perspectives, such as the model architecture, training strategy, and dataset.

Firstly, a large number of samples for model training have already been verified by many studies. However, setting up such a kind of sufficient and diversified is a difficult, boring, and time-consuming task most of the time. To set up the synthetic dataset, only a few labels are needed to generate a number of images with the corresponding labels, which greatly reduce manual operation. The model trained on a synthetic with good performance verified that the image generated by this synthetic method was very close to the real environment. This method also generated augmented images of data as good supplements. The model trained on O-S was much better than the model trained on the other two datasets. This result indicated that the generated images had differences with conventional captured images. Thus, the synthetic and conventional complemented each other. Therefore, O-S had more diversity than the other two datasets. From the results, we can confidently believe that such kind synthetic dataset is effective for the fruit detection task.

Secondly, by adding the center loss function to the architecture of the model, an attempt was made to reduce variation and enhance inter-class variation and hence reduce some false-positive and false-negative results. Although center loss had certain advantages, it was very hard to increase the weight of this loss function. When we increase the weight of center loss more than 0.5, this whole architecture was very hard to converge. Even worse, the accuracy reduced rapidly. It is due to the loss function still plays an irreplaceable key role in this detection task.

Thirdly, the OHEM strategy was used during training to strengthen the performance of the detector by retraining the hard examples. Such a training strategy can improve the performance to a certain extent. However, it is difficult to improve the detector performance greatly due to the limitation of hard example diversity.

Finally, the model was evaluated on three levels of occlusion severity. The experiments showed that heavy occlusion decreases the accuracy of prediction results. Mainly, there are two reasons for this. One is that the obscured objects destroy integrity. It is liable to predict erroneous results if only a few parts of the object are visible. The other reason is the non-maximum suppression (NMS) strategy. At present, most of the object detection task is based on deep learning NMS methods to select the best candidate bounding box during training or testing. However, when there are dense fruits on the tree, one fruit prone to be heavily occluded by the others, the NMS method will suppress the candidate bounding box, which is not the max probability value but has a high value of IOU with the max probability value's candidate bounding box. So, there is a great possibility that such objects will be missed by the detector.

7. Conclusion and Future Work

This study presents further improved coarse-to-fine networks for fruit detection in orchards, from three aspects. First, for increasing the diversity samples for the model training and reducing the cost of manual collection and labeling, synthetic images were generated to set up

a dataset for training deep CNNs. The experimental results confirm that training based on this synthetic dataset greatly improved the accuracy of the detector. Then, for better distinguishing features of objects and background, the loss function was modified by adding the center-loss function to improve the discrimination power. Furthermore, for intensifying the learning effect of the model, the OHEM strategy was employed during training. By re-learning hard examples, the model provided better performance for the prone to be false predicted images. Finally, the output results reach 1.16 frames per second which show the presented method could be promoted in the practical agricultural field.

Moreover, intensive and extensive discussions about the probable reasons caused by the severity of occlusion. In further, the detection task for heavily occluded objects would be the focus of the study.

References

[1] A. Durand-Petiteville, S. Vougioukas, and D. C. Slaughter, "Real-time segmentation of strawberry flesh and calyx from images of singulated strawberries during postharvest processing," *Computers and Electronics in Agriculturre*, vol. 142, pp. 298-313, 2017. Article (CrossRef Link)

[2] W. Mao, B. Ji, J. Zhan, X. Zhang, and X. Hu, "Apple location method for the apple harvesting robot," in *Proc. of the 2nd International Congress on Image and Signal Processing*, pp. 1-5, 2009. Article (CrossRef Link)

[3] N. Behroozi-Khazaei and M. R. Maleki, "A robust algorithm based on color features for grape cluster segmentation," *Computers and Electronics in Agriculturre*, vol. 142, pp. 41-49, 2017. Article (CrossRef Link)

[4] J. Ma, K. Du, L. Zhang, F. Zheng, J. Chu, and Z. Sun, "A segmentation method for greenhouse vegetable foliar disease spots images using color information and region growing," *Computers and Electronics in Agriculture*, vol. 142, pp. 110-117, 2017. Article (CrossRef Link)

[5] J. Lu, J. Hu, G. Zhao, F. Mei, and C. Zhang, "An in-field automatic wheat disease diagnosis system," *Computers and Electronics in Agriculture*, vol. 142, pp. 369-379, 2017. Article (CrossRef Link)

[6] A. Mohapatra, S. Shanmugasundaram, and R. Malmathanraj, "Grading of ripening stages of red banana using dielectric properties changes and image processing approach," *Computers and Electronics in Agriculture*, vol. 143, pp. 100–110, 2017. Article (CrossRef Link)

[7] L. M. Azizah, S. F. Umayah, S. Riyadi, C. Damarjati, and N. A. Utama, "Deep learning implementation using convolutional neural network in mangosteen surface defect detection," in *Proc. of the 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pp. 242-246, 2017. Article (CrossRef Link)

[8] A. Gongal, S. Amatya, M. Karkee, Q. Zhang, and K. Lewis, "Sensors and systems for fruit detection and localization: A review," *Computers and Electronics in Agriculture*, vol. 116, pp. 8-19, 2015. Article (CrossRef Link)

[9] J. Lu, W. S. Lee, H. Gan, and X. Hu, "Immature citrus fruit detection based on local binary pattern feature and hierarchical contour analysis," *Biosystems Engineering*, vol. 171, pp. 78-90, 2018. Article (CrossRef Link)

[10] Y. Shi, W. Huang, J. Luo, L. Huang, and X. Zhou, "Detection and discrimination of pests and diseases in winter wheat based on spectral indices and kernel discriminant analysis," *Computers and Electronics in Agriculture*, vol. 141, pp. 171-180, 2017. Article (CrossRef Link)

[11] L. Zhang, G. Gui, A. M. Khattak, M. Wang, W. Gao, and J. Jia, "Multi-task cascaded convolutional networks based intelligent fruit detection for designing automated robot," *IEEE Access*, vol. 7, pp. 56028-56038, 2019. Article (CrossRef Link)

[12] X. D. Bai, Z. G. Cao, Y. Wang, Z. H. Yu, X. F. Zhang, and C. N. Li, "Crop segmentation from images by morphology modeling in the CIE L* a* b* color space," *Computers and Electronics in Agriculture*, vol. 99, pp. 21-34, 2013. Article (CrossRef Link)

[13] M. Nagle, K. Intani, G. Romano, B. Mahayothee, V. Sardsud, and J. Müller, "Determination of surface color of 'all yellow'mango cultivars using computer vision," *International Journal of Agricultural and Biological Engineering*, vol. 9, no. 1, pp. 42-50, 2016. Article (CrossRef Link)
[14] 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. Article (CrossRef Link)
[15] A. Yadav, N. Sengar, A. Issac, and M. K. Dutta, "Image processing based acrylamide detection from fried potato chip images using continuous wavelet transform," *Computers and Electronics in Agriculture*, vol. 145, pp. 349-362, 2018. Article (CrossRef Link)
[16] M. P. Arakeri and Lakshmana, "Computer vision based fruit grading system for quality evaluation of tomato in agriculture industry," *Procedia Computer Science*, vol. 79, pp. 426-433, 2016. Article (CrossRef Link)
[17] S. Siddesha, S. K. Niranjan, and V. N. M. Aradhya, "Segmentation of coconut crop bunch from tree images," in *Proc. of 2016 Second International Conference on Cognitive Computing and Information Processing (CCIP)*, pp. 1-6, 2016. Article (CrossRef Link)
[18] R. Hassankhani and H. Navid, "Potato sorting based on size and color in machine vision system," *Journal of Agricultural Science*, vol. 4, no. 5, pp. 235-244, 2012. Article (CrossRef Link)
[19] Y. Shao, "Supervised global-locality preserving projection for plant leaf recognition," *Computers and Electronics in Agriculture*, vol. 158, pp. 102-108, 2019. Article (CrossRef Link)
[20] D. Pise and G. D. Upadhye, "Grading of harvested mangoes quality and maturity based on machine learning techniques," in *Proc. of 2018 International Conference on Smart City and Emerging Technology (ICSCET)*, 2018, pp. 1-6. Article (CrossRef Link)
[21] W. Xu, S. Lee, and E. Lee, "A Robust Method for Partially Occluded Face Recognition," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 7, pp. 2667-2682, 2015. Article (CrossRef Link)
[22] J. Liu, J. Tan, J. Qin, and X. Xiang, "Smoke Image Recognition Method Based on the optimization of SVM parameters with Improved Fruit Fly Algorithm," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 8, pp. 3534-3549, 2020. Article (CrossRef Link)
[23] W. Ahmad, S. M. A. Shah, and A. Irtaza, "Plants Disease Phenotyping using Quinary Patterns as Texture Descriptor," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 8, pp. 3312-3327, 2020. Article (CrossRef Link)
[24] K. Yu, L. Lin, M. Alazab, L. Tan, and B. Gu, "Deep Learning-Based Traffic Safety Solution for a Mixture of Autonomous and Manual Vehicles in a 5G-Enabled Intelligent Transportation System," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1-11, 2020. Article (CrossRef Link)
[25] Z. Guo, Y. Shen, A. K. Bashir, M. Imran, N. Kumar, D. Zhang, and K. Yu, "Robust Spammer Detection Using Collaborative Neural Network in Internet of Thing Applications," *IEEE Internet of Things Journal*, early access. Article (CrossRef Link)
[26] Z. Guo, K. Yu, Y. Li, G. Srivastava, and J. C. W. Lin, "Deep Learning-Embedded Social Internet of Things for Ambiguity-Aware Social Recommendations," *IEEE Transactions on Network Science and Engineering*, early access. Article (CrossRef Link)
[27] Y. Wang, G. Gui, T. Ohtsuki, and F. Adachi, "Multi-task learning for generalized automatic modulation classification under non-Gaussian noise with varying SNR conditions," *IEEE Transactionsa on Wireless Commuications*, early access. Article (CrossRef Link)
[28] G. Gui, M. Liu, F. Tang, N. Kato, and F. Adachi, "6G: Opening new horizons for integration of comfort, security and intelligence," *IEEE Wireless Communications*, vol. 27, no. 5, pp. 126-132, Oct. 2020. Article (CrossRef Link)
[29] Y. Lin, M. Wang, X. Zhou, G. Ding, and S. Mao, "Dynamic spectrum interaction of UAV flight formation communication with priority: A deep reinforcement learning approach," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 3, pp. 892-903, 2020. Article (CrossRef Link)
[30] Y. Lin, Y. Tu, Z. Dou, L. Chen, and S. Mao, "Contour stella image and deep learning for signal recognition in the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, early access. Article (CrossRef Link)

[31] T. Nishi, S. Kurogi, and K. Matsuo, "Grading fruits and vegetables using RGB-D images and convolutional neural network," in *Proc. of 2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1-6, 2017. Article (CrossRef Link)
[32] L. Hou, Q. Wu, Q. Sun, H. Yang, and P. Li, "Fruit recognition based on convolution neural network," in *Proc. of the 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 18-22, 2016. Article (CrossRef Link)
[33] J. Zhang, L. He, M. Karkee, Q. Zhang, X. Zhang, and Z. Gao, "Branch detection for apple trees trained in fruiting wall architecture using depth features and Regions-Convolutional Neural Network (R-CNN)," *Computers and Electronics in Agriculture*, vol. 155, pp. 386-393, 2018. Article (CrossRef Link)
[34] S. A. Akbar, S. Chattopadhyay, N. M. Elfiky, and A. Kak, "A novel benchmark RGBD dataset for dormant apple trees and its application to automatic pruning," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 347-354, 2016. Article (CrossRef Link)
[35] L. Zhang, J. Jia, Y. Li, W. Gao, and M. Wang, "Deep Learning based Rapid Diagnosis System for Identifying Tomato Nutrition Disorders," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 4, pp. 2012-2027, 2019. Article (CrossRef Link)
[36] G. Zeng, "Fruit and vegetables classification system using image saliency and convolutional neural network," in *Proc. of* the *3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, pp. 613-617, 2017. Article (CrossRef Link)
[37] Z. M. Khaing, Y. Naung, and P. H. Htut, "Development of control system for fruit classification based on convolutional neural network," in *Proc. of IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp. 1805-1807, 2018. Article (CrossRef Link)
[38] L. Zhang, J. Jia, G. Gui, X. Hao, W. Gao, and M. Wang, "Deep learning based improved classification system for designing tomato harvesting robot," *IEEE Access*, vol. 6, pp. 67940-67950, 2018. Article (CrossRef Link)
[39] 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. Article (CrossRef Link)
[40] X. Liu, S. W. chen, S. Aditya, N. Sivakumar, S. Dcunha, C. Qu, C. J. Taylor, J. Das, and V. Kumar, "Robust fruit counting: Combining deep learning, tracking, and structure from motion," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1045-1052, 2018. Article (CrossRef Link)
[41] 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 Intellignece*, vol. 39, no. 6, pp. 1137-1149, 2017. Article (CrossRef Link)
[42] S. Tu, Y. Xue, C. Zheng, Y. Qi, H. Wan, and L. Mao, "Detection of passion fruits and maturity classification using Red-Green-Blue Depth images," *Biosystems Engineering*, vol. 175, pp. 156-167, 2018. Article (CrossRef Link)
[43] H. Gan, W. S. Lee, V. Alchanatis, R. Ehsani, and J. K. Schueller, "Immature green citrus fruit detection using color and thermal images," *Computers and Electronics in Agriculture*, vol. 152, pp. 117-125, 2018. Article (CrossRef Link)
[44] 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, pp. 1-23, 2016. Article (CrossRef Link)
[45] S. Bargoti and J. Underwood, "Deep fruit detection in orchards," in *Proc. of 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3626-3633, 2017. Article (CrossRef Link)
[46] J. Gené-Mola, V. Vilaplana, J. R. Rosell-Polo, J. R. Morros, J. Ruiz-Hidalgo, and E. Gregorio, "KFuji RGB-DS database: Fuji apple multi-modal images for fruit detection with color, depth and range-corrected IR data," *Data in Brief*, vol. 25, 2019. Article (CrossRef Link)
[47] L. Fu, Y. Feng, Y. Majeed, X. Zhang, J. Zhang, M. Karkee, and Q. Zhang, "Kiwifruit detection in field images using Faster R-CNN with ZFNet," *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 45-50, 2018. Article (CrossRef Link)

[48] S. Tu, J. Pang, H. Liu, N. Zhuang, Y. Chen, C. Zheng, H. Wan, and Y. Xue, "Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images," *Precision Agriculture*, pp. 1-20, 2020. Article (CrossRef Link)
[49] N. Häni, P. Roy, and V. Isler, "Minneapple: A benchmark dataset for apple detection and segmentation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 852-858, 2020. Article (CrossRef Link)
[50] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779-788, 2016. Article (CrossRef Link)
[51] J. R. A. Farhadi and J. Redmon, "YOLOv3: An incremental improvement," *arXiv:1804.02767*, 2018. Article (CrossRef Link)
[52] K. Bresilla, G. D. Perulli, A. Boini, B. Morandi, L. C. Grappadelli, and L. Manfrini, "Single-shot convolution neural networks for real-time fruit detection within the tree," *Frontier Plant Science*, vol. 10, 2019. Article (CrossRef Link)
[53] 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 Electronics in Agriculture*, vol. 157, pp. 417-426, 2019. Article (CrossRef Link)
[54] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. of 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248-255, 2009. Article (CrossRef Link)
[55] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *Proc. of IEEE International Conference on Computer Vision*, pp. 1310-1319, 2017. Article (CrossRef Link)
[56] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut' interactive foreground extraction using iterated graph cuts," *ACM Transations Graphics*, vol. 23, no. 3, pp. 309-314, 2004. Article (CrossRef Link)
[57] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. of European Conference on Computer Vision*, pp. 499-515, 2016. Article (CrossRef Link)
[58] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc.of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 761-769, 2016. Article (CrossRef Link)

Li Zhang is a Ph.D. candidate at the department of the college of information and electrical engineering, China Agricultural University, Beijing, China. She researches on deep learning-based object recognition, tracking, and detection for the vision system of agricultural robots.

Yanzhao Ren, Post Doctorate. He obtained his Master of Engineering in Agricultural Machine Engineering from Shandong University of Technology, Zibo, China in 2012. He is currently pursuing his Ph.D. in Agricultural Electrification and Automation at China Agricultural University, Beijing. His research direction is computer application technology and intelligent information processing. His research subjects include agricultural IoT, automatic operating equipment, intelligent video projection equipment.

Sha Tao received the Ph.D. degree from the College of Food Science & Nutritional Engineering, China Agricultural University, in 2013. From 2014 to 2016, she was a Post-Doctoral Research Fellow with the Key Laboratory of Agricultural Informatization Standardization, Ministry of Agriculture (Prof. Wanlin Gao Laboratory), College of Information and Electrical Engineering, China Agricultural University, where she is currently a lecturer in this college. Her research mainly focuses on the processing and storage of agricultural products technologies.

Jingdun Jia is a professor at the department of the college of information and electrical engineering, China Agricultural University, Beijing, China. He is also the director National Rural Technology Development Center, Ministry of Science and Technology. He has been the host and attends a variety of internationally renowned conferences. And he also participated in the discussion of strategic planning of several major national projects.

Wanlin Gao is a professor of the College of Information and Electrical Engineering of China Agricultural University. He is also a member of the Science and Technology Committee of the Ministry of Agriculture, the member of the Agriculture and Forestry Committee of Computer Basic Education in Colleges and Universities, a senior member of the Society of Chinese Agricultural Engineering, etc. He received degrees (B.S., 1990; S.M., 2000; Ph.D., 2010) from China Agricultural University, and has worked in China Agricultural University since 1990. His major research area is the informatization of new rural areas, intelligence agriculture, and the service for rural comprehensive information.