Computers and Electronics in Agriculture 157 (2019) 417–426

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Apple detection during different growth stages in orchards using the improved YOLO-V3 model

Yunong Tian, Guodong Yang, Zhe Wang, Hao Wang, En Li*, Zize Liang

Institute of Automation, Chinese Academy of Sciences, No. 95 Zhongguancun East Road, Beijing 100190, China
University of Chinese Academy of Sciences, No.19(A) Yuquan Road, Beijing 100049, China

ARTICLE INFO

Keywords:
Apple images acquisition
Image augmentation
Deep learning
YOLOV3-dense
Real-time detection

ABSTRACT

Real-time detection of apples in orchards is one of the most important methods for judging growth stages of apples and estimating yield. The size, colour, cluster density, and other growth characteristics of apples change as they grow. Traditional detection methods can only detect apples during a particular growth stage, but these methods cannot be adapted to different growth stages using the same model. We propose an improved YOLO-V3 model for detecting apples during different growth stages in orchards with fluctuating illumination, complex backgrounds, overlapping apples, and branches and leaves. Images of young apples, expanding apples, and ripe apples are initially collected. These images are subsequently augmented using rotation transformation, colour balance transformation, brightness transformation, and blur processing. The augmented images are used to create training sets. The DenseNet method is used to process feature layers with low resolution in the YOLO-V3 network. This effectively enhances feature propagation, promotes feature reuse, and improves network performance. After training the model, the performance of the trained model is tested on a test dataset. The test results show that the proposed YOLOV3-dense model is superior to the original YOLO-V3 model and the Faster R-CNN with VGG16 net model, which is the state-of-art fruit detection model. The average detection time of the model is 0.304 s per frame at 3000 × 3000 resolution, which can provide real-time detection of apples in orchards. Moreover, the YOLOV3-dense model can effectively provide apple detection under overlapping apples and occlusion conditions, and can be applied in the actual environment of orchards.

1. Introduction and related works

Nowadays, labour in farms and orchards primarily relies on skilled farmers. Manual work consumes time and increases production costs, and workers that lack knowledge and experience make unnecessary mistakes. With the developments in precision agriculture and information technology, crop imaging has become an important means of gathering crop growth information (Zhao et al., 2016). Intelligent agriculture has become a popular concept (Tyagi, 2016) and image information can be used to accurately judge crop growth and estimate crop yield (Wang et al., 2013). The automation of agricultural production also makes it possible to continuously monitor crop growth and nutrition status, in order to carry out independent agricultural management and control.

Fluctuating illumination, complex background, dense fruit distribution, overlapping fruit, branches and leaves, the camera's viewing angle, distance, and other factors in orchards can have certain impacts on target detection. Many researchers have provided and improved different algorithms for crop detection and localization. Hamuda et al. (2018) used Kalman filtering and the Hungarian algorithm to detect crops in the field. These experiments were conducted without overlapping crops. The background in the images was soil, which is relatively simple, thus the method is not suitable for detecting densely distributed fruit with occlusion. Lu and Sang (2015) proposed a method for citrus fruit recognition under varying canopy illumination based on colour and contour information. This method could adapt to the natural environment with complex illumination and background, but the detection performance is poor when the citrus is small in the image. Linker et al. (2012) proposed a method for detecting apples in natural lighting. This method uses colour and smoothness to detect a set of pixels with high probability belonging to apples and form a "seed area". The method then determines whether the region contains an apple according to the coincidence ratio between the "seed area" and an apple model. This method can effectively detect a region that contains an apple, but it will produce large error in the case of dense distribution and large overlap of apples.

* Corresponding author.
E-mail address: en.li@ia.ac.cn (E. Li).

https://doi.org/10.1016/j.compag.2019.01.012
Received 16 October 2018; Received in revised form 11 January 2019; Accepted 12 January 2019
Available online 19 January 2019

With the development of machine learning, deep learning technology has been widely used in agriculture (Kamilaris and Prenafeta-Boldú, 2018). Deep learning can be used for crop classification (Lee et al., 2017; Tang et al., 2017; Zhang et al., 2016), crop image segmentation (Arribas et al., 2011; Dias et al., 2018), crop target detection (Bargoti and Underwood, 2016; Yamamoto et al., 2014), and other tasks (Rahnemoonfar and Sheppard, 2017). Crop classification is the basis of crop detection. Zhang et al. (2017) designed a 13-layer convolutional neural network (CNN) for fruit classification with accuracy of 94.94%. This algorithm is the state-of-art method in fruit category classification. Target detection refers to category classification and target localization in an image. Image segmentation based on deep learning is one of the target detection methods. The number of target areas in the image can be calculated and their locations can be obtained through target area segmentation. Chen et al. (2017) used blob detectors based on fully connected CNNs to extract candidate regions in the image, segment object areas, and calculate the number of fruits using a subsequent CNN counting algorithm. Dyrmann et al. (2017) used a fully-connected CNN to detect weeds automatically when many leaves in the image were blocked. In order to segment the target area more accurately in a complex natural environment, Dias et al. (2018) used CNN and support vector machine (SVM) methods to extract the characteristics of apple blossoms automatically against a complex background; the method produced relatively accurate apple blossom area segmentation results. Image segmentation methods based on deep learning have produced good results in crop area segmentation. However, these methods cannot accurately segment the regions of each target in crops with serious overlap. Faster R-CNN (Ren et al., 2016) uses the region proposal network (RPN) method to detect a region of interest (RoI) in the image. Then a classifier is used to classify bounding boxes, and fine tuning is used to process the bounding boxes. Finally, the target can be detected accurately. It provides guiding significance for crop detection, crop yield estimation, crop growth judgement and agricultural management. Bargoti and Underwood (2016), Inkyu et al. (2016) used the Faster R-CNN method to detect a variety of fruits and produced good results. Faster R-CNN with VGG16 net (Simonyan and Zisserman, 2014) is the state-of-art method in fruit detection (Kamilaris and Prenafeta-Boldú, 2018). However, Faster R-CNN consists of two parts: region proposal networks (RPN) and classification networks, thus the detection speed is slow and cannot produce real-time results with high image resolution.

The You Only Look Once (YOLO) method (Redmon et al., 2016; Redmon and Farhadi, 2017, 2018) unifies target classification and localization into a regression problem. A YOLO network does not require RPN, and it directly performs regression to detect targets in the image. The network provides much faster detection. The state-of-art version (YOLO-V3) not only has high detection accuracy and speed, but also performs well with detecting small targets. However, the YOLO-V3 model has not been widely used for fruit detection.

Feature maps are gradually shrinking due to the use of convolution and down-sampling operations in deep neural networks. The DenseNet architecture (Huang et al., 2017) is proposed for using the input features of neural networks more efficiently. In the DenseNet architecture, each layer uses the feature maps from all preceding layers as inputs, and its own feature map is used as the input of all subsequent layers. These feature maps are connected by depth concatenation. The basic structure of DenseNet primarily consists of two components: a dense block and a transition layer. The dense block is a group of densely connected feature maps. The layer between two adjacent dense blocks is referred to as the transition layer and change feature map sizes via convolution and pooling. The application of DenseNet in neural network strengthens feature propagation, which can effectively solve the vanishing gradient problem and improve the classification accuracy of neural networks.

While viewing images of apples, the illumination conditions in orchards are inconstant, the background is complex, the camera's viewing distance is not fixed, the apples are densely distributed and overlap, and ubiquitous branches and leaves shelter fruit. All these problems present great challenges to detecting apples in orchards. The sizes, colours, and cluster densities of apples in various growth stages are also different. Apples are small, green, and densely clustered when they are still young. During the expansion period, the volume of an apple becomes larger, its colour changes, and the cluster density decreases due to pruning and other agricultural activities. In the ripe stage, apples are large, are usually red or reddish yellow, and are sparsely distributed. Traditional methods are not suitable for detecting apples during different growth stages in complex and changing environments. There is also a trade-off between the accuracy and real-time performance of deep learning methods. In order to better solve these problems, the state-of-art YOLO-V3 algorithm (Redmon and Farhadi, 2018) is used for detecting apples in real-time in this study. In order to improve the detection performance of the YOLO-V3 network, DenseNet is used to optimize feature layers with low resolution. Images of apples in the main growth stages, including the young, expanding, and ripe stages, are collected and used as input data for training the neural network. The trained neural network is used for detecting apples and identify their growth stages.

The rest of the paper is organized as follows. Section 2 introduces the methods for pre-processing the image dataset, including image acquisition, image data augmentation, and the creation of image datasets. Section 3 introduces the improved YOLO-V3 algorithm, which incorporates the DenseNet method. Section 4 introduces the relevant experiments and a discussion of the experimental results. Finally, the conclusions and prospects of this paper are described.

2. Image data pre-processing

2.1. Image data acquisition

In this study, image acquisition was conducted using a camera with 3000 × 4000 pixel resolution during different growth stages. The orchards are located in Lingbao, Henan, China.

The image data used in this paper were collected in apple orchards during cloudy and sunny weather conditions. The collection periods included 8 a.m., 1 p.m., and 5: 30 p.m. The illumination conditions included front-lighting, backlighting, side-lighting, and scattered lighting. During image acquisition, the camera's viewing direction was set parallel to the sunlight illumination direction in order to simulate front-lighting. The camera was aimed antiparallel to the sunlight illumination direction to simulate backlighting. The camera was aimed perpendicular to the sunlight illumination direction to simulate side-lighting. Images were also gathered in cloudy conditions to simulate scattered lighting.

320 images of apples were collected from orchards during each of the three growth stages. Half of the apple images during each growth stage were randomly selected for use in the training set. Considering that the camera's viewing angles will affect the detection performance, some images were collected from multiple viewing angles during image acquisition. Among these selected 480 images, 94 images were collected while changing the viewing angle, including images of 30 young apples, 32 expanding apples, and 32 ripe apples.

These 480 images were then expanded to 4800 images using data augmentation methods, yielding the training dataset. The training dataset is used to train the detection model. The remaining 480 images are used as the test dataset to verify the detection performance of the YOLOV3-dense model.

2.2. Image data augmentation

Apples in orchards were detected and the growth stages of apples were judged. Since the angle and intensity of sunlight illumination varies greatly during the day, whether the neural network can process the images collected at different time of the day depends on the

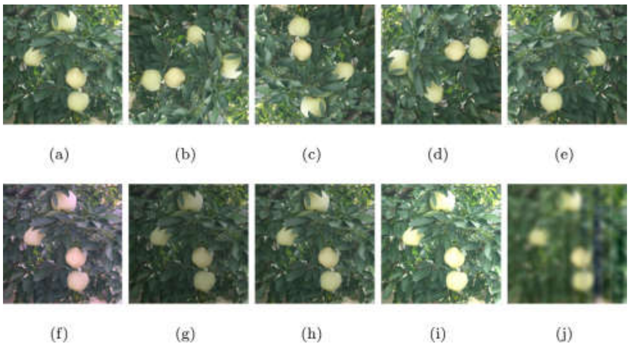


Fig. 1. Image augmentation methods: (a) original image, (b) 90° clockwise rotation, (c) 180° clockwise rotation, (d) 270° clockwise rotation, (e) horizontal mirror, (f) colour balance processing, (g-i) brightness transformation, and (j) blur processing.

integrity of the training dataset. In order to enhance the richness of the experimental dataset, the collected images were pre-processed in terms of colour, brightness, rotation, and image definition, and the dataset was augmented as shown in Fig. 1.

2.2.1. Data augmentation: image colour

The human visual system can determine colour invariance of the surface of an object under changing light and imaging conditions, but imaging devices do not have such colour invariance. Different lighting conditions will lead to a certain deviation between the image colour and real colour. The gray world algorithm (Lam, 2005) was used to eliminate the influence of lighting on colour rendering. The gray world algorithm is based on the gray-world hypothesis, which holds that the average values of R, G, and B components tend to the same grey value for an image exhibiting a large number of colour changes. Physically, the gray world algorithm assumes that the average reflection of light from natural objects is generally a fixed value, which is approximately gray. The colour balance algorithm was used to apply this hypothesis to the images in the training set. The influence of ambient light can be eliminated from the image, yielding the original image.

2.2.2. Data augmentation: image brightness

The brightness of images in the training set was processed as follows. Three values were randomly selected from l_{min} to l_{max} and were used to adjust the brightness of the original images, and the three new results were added to the training set. If the image brightness is too high or too low, bounding boxes will be difficult to draw during manual annotation because the edge of the target is unclear. During training, these training set images will have a detrimental influence on the performance of the detection model. In order to avoid generating such images, an appropriate range of image brightness transformations was selected depending on whether the target edge can be accurately identified during manual annotation, i.e., $l_{min} = 0.6$ and $l_{max} = 1.4$. This method can simulate the situation of orchards under different illumination intensities. These values compensate for shortcomings of the neural network, which is not robust to various illumination intensities caused by the concentrated time of image acquisition.

2.2.3. Data augmentation: image rotation

To further expand the image dataset, the original images were rotated by 90°, 180°, and 270° and mirrored. The rotated images can also improve the detection performance of the neural network.

2.2.4. Data augmentation: image definition

The acquired images may not be clear due to the camera's long viewing distance, incorrect focus, or camera movement. Indistinct images can also affect the detection results of the neural network. Therefore, in this paper, images augmented by colour, brightness, and rotation were randomly blurred to simulate indistinct images. The robustness of the detection model will be further enhanced by using indistinct images as samples.

2.3. Images annotation and dataset production

In order to better compare the performance of different algorithms, images in the training set were converted to PASCAL VOC format. The lengths of the training set images were rescaled to 500 pixels and the widths were adjusted accordingly to maintain the original aspect ratio while creating the training set. Manual annotation was applied after the images were numbered. Bounding boxes were drawn and the categories were classified manually. Positive samples with insufficient or unclear pixel area were not labelled to prevent over-fitting in the neural network. In the case of occlusion, a target whose occlusion area was greater than 85% and the target at the edge of the image with less than 15% area were not labelled. The completed dataset is shown in Table 1.

3. Methodologies

3.1. YOLO-V3

The YOLO-V3 (Redmon and Farhadi, 2018) network is evolved from the YOLO (Redmon et al., 2016) and YOLO-V2 (Redmon and Farhadi, 2017) networks. Compared with the Faster R-CNN network, the YOLO network transforms the detection problem into a regression problem. It does not require a proposal region, and it generates bounding box coordinates and probabilities of each class directly through regression. This greatly increases the detection speed compared to Faster R-CNN.

The YOLO detection model is shown in Fig. 2. The network divides each image in the training set into $S \times S$ ($S = 7$) grids. If the center of the target ground truth falls in a grid, then the grid is responsible for detecting the target. Each grid predicts B bounding boxes and their confidence scores, as well as C class conditional probabilities. *Confidence* is defined as follows:

$$Confidence = p_r(Object) \times IoU_{pred}^{truth},\ p_r(Object) \in \{0, 1\} \tag{1}$$

When the target is in the grid, $p_r(Object) = 1$ and 0 otherwise. IoU_{pred}^{truth} is used to denote the coincidence between the reference and the predicted bounding box. The confidence reflects whether the grid contains objects and the accuracy of the predicted bounding box when it contains objects. When multiple bounding boxes detect the same target, YOLO uses the non-maximum suppression (NMS) method to select the best bounding box.

Although YOLO provides greater speed compared with Faster R-CNN, it has a large detection error. In order to solve this problem, YOLO-V2 introduces the idea of the "anchor box" in Faster R-CNN and uses k-means clustering method to generate suitable priori bounding boxes. Thus, the number of anchor boxes required to achieve the same

Table 1
The number of images generated by data augmentation methods.

	Original data	Color	Brightness	Rotation	Definition	Total
Number of young apple images	160	160	480	640	160	1600
Number of expanding apple images	160	160	480	640	160	1600
Number of ripe apple images	160	160	480	640	160	1600

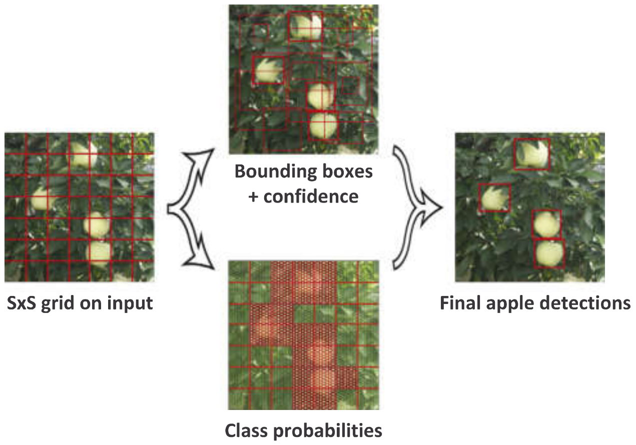


Fig. 2. YOLO Detection.

intersection over union (IoU) results decreases. YOLO-V2 improves the network structure and uses a convolution layer to replace the fully connected layer in the output layer of YOLO. YOLO-V2 also introduces batch normalization, a high resolution classifier, dimension clusters, direct location prediction, fine-grained features, multi-scale training, and other methods that greatly improve the detection accuracy compared with YOLO.

YOLO-V3 is an improved version of YOLO-V2. It uses multi-scale prediction to detect the final target, and its network structure is more complex than YOLO-V2. YOLO-V3 predicts bounding boxes on different scales, and multi-scale prediction makes YOLO-V3 more effective for detecting small targets than YOLO-V2.

3.2. Densely connected neural networks

The feature maps are reduced while training the neural network due to convolution and down-sampling, and the feature information is lost during transmission. DenseNet was proposed to make more effective use of feature information (Huang et al., 2017). It connects each layer to other layers in feedforward mode, thus layer l receives all the feature maps of the preceding layers $x_0, x_1, \ldots, x_{l-1}$ as input.

$$x_l = H_l[x_0, x_1, \ldots, x_{l-1}] \tag{2}$$

where $[x_0, x_1, \ldots, x_{l-1}]$ is a splice of the feature maps of layers $x_0, x_1, \ldots, x_{l-1}$, and H_l is a function used to process the spliced feature maps. This allows DenseNet to mitigate gradient vanishing, enhance feature propagation, facilitate feature reuse, and greatly reduce the number of parameters.

3.3. The proposed algorithm

Fig. 3 shows how the Darknet-53 architecture of YOLO-V3 is used as the basic network architecture, and DenseNet is used instead of the original transfer layers with lower resolution to enhance feature propagation and facilitate feature reuse and fusion.

The specific network parameters of YOLOV3-dense are shown in Fig. 4. In order to better process high resolution images, the input image is first resized to 512 × 512 pixels, replacing the original images with 256 × 256 pixels. Then the 32 × 32 and 16 × 16 down-sampling layers in the improved network are replaced by the DenseNet structure. In this paper, the transfer function H_l uses the function BN-ReLU-Conv(1 × 1)-BN-ReLU-Conv(3 × 3), which is a combination of batch normalization (BN), rectified linear units (ReLU), and convolution (Conv). H_l provides nonlinear transformation of $x_0, x_1, \ldots, x_{l-1}$ layers. x_i consists of 64 feature layers, each with 32 × 32 resolution. H_1 applies BN-ReLU-Conv (1 × 1) nonlinear operation on x_0, and then performs BN-ReLU-Conv (3 × 3) operation on the result. H_2 applies the same operation on the feature map formed by $[x_0, x_1]$. The result x_2 and $[x_0, x_1]$ are spliced into $[x_0, x_1, x_2]$ and used as the input to H_3. The result x_3 and $[x_0, x_1, x_2]$ are spliced into $[x_0, x_1, x_2, x_3]$, as the input of H_4. Eventually, the feature layer $[x_0, x_1, x_2, x_3, x_4]$ continues to propagate forward. In the layers with 16 × 16 resolution, feature propagation and feature layer splicing are also performed as mentioned above. Finally, the feature layer is spliced into 16 × 16 × 1024 and is propagated forward.

During training, when the image features are transferred to the lower resolution layers, the latter feature layer will receive the features of all the feature layers in front of it in DenseNet, thus reducing feature loss. In this way, features can be reused between convolution layers with low resolution; the feature usage rate increases and the usage effect of features improves.

Finally, the YOLOV3-dense model proposed in this paper predicts bounding boxes at three different scales: 64 × 64, 32 × 32, and 16 × 16. It also classifies target categories to provide apple detection.

4. Experiments and discussion

The YOLOV3-dense detection model used in this study was modified using the Darknet framework (Redmon and Farhadi, 2018). The detection models were trained and tested on an NVIDIA Tesla V100 server. The network initialization parameters are shown in Table 2.

In order to improve the detection accuracy of the model and to adapt the input required for the Darknet framework, the input images were adjusted to 512 × 512 pixels. Taking into account the memory constraints of the server, the batch size was set to 8 in this paper. 70,000 training steps were used in order to better analyse the training process. Parameters such as momentum, initial learning rate, weight decay regularization, and other parameters referred to the original parameters in the YOLO-V3 model. The model was trained after defining the training parameters. The learning rate decreased to 0.0001 after 40, 000 steps and to 0.00001 after 50, 000 steps.

In this paper, a series of experiments with the trained YOLOV3-dense model were conducted with the test images to verify the performance of the algorithm. Images with 3000 × 3000 resolution were used for testing. The related indicators for evaluating the effectiveness of the neural network models are as follows:

A. Precision, Recall, and F1 Score

For binary classification problems, samples can be divided into four types: true positive (TP), false positive (FP), true negative (TN), and false negative (FN), according to the combinations of the true class and predicted class of the learner. The confusion matrix for the classification results is shown in Table 3.

Precision(P) and recall(R) are defined as follows:

$$P = \frac{TP}{TP + FP} \tag{3}$$

$$R = \frac{TP}{TP + FN} \tag{4}$$

The precision-recall curve, called P-R curve for short, can be obtained by using the precision ratio as vertical axis and the recall ratio as the horizontal axis. The F1 score was also used to evaluate the performance of the model. The definition of the F1 score is shown as follows:

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{5}$$

B. Loss Function

Loss function is one criterion for evaluating the performance of a model. The loss function in YOLO is defined as follows:

$$Loss = Error_{coord} + Error_{iou} + Error_{cls} \tag{6}$$

The coordinate prediction error $Error_{coord}$ is defined as follows:

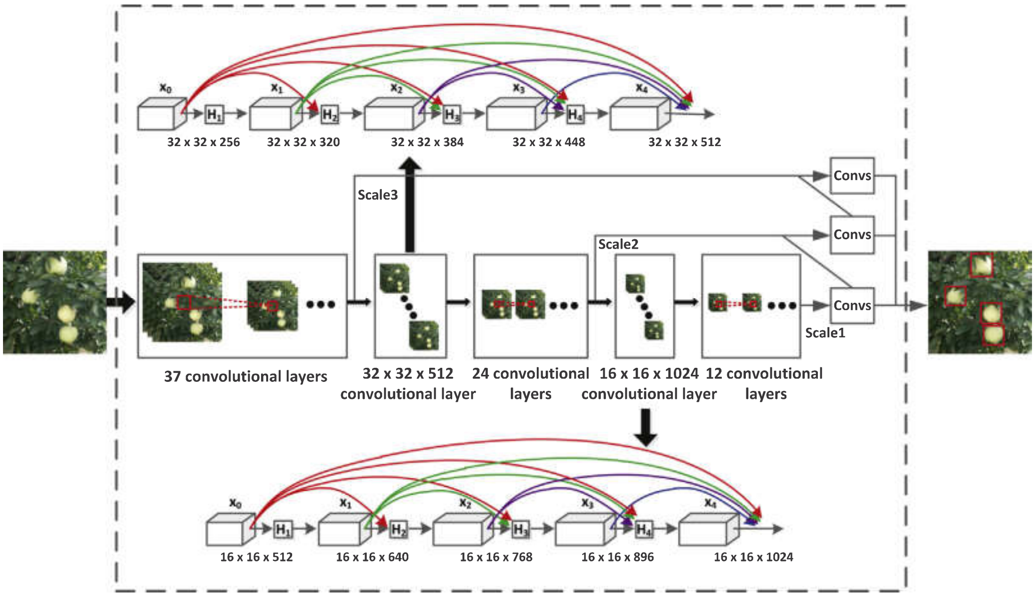


Fig. 3. YOLOV3-dense network structure diagram.

$$Error_{coord} = \lambda_{coord}\Sigma_{i=1}^{S^2}\Sigma_{j=1}^{B}\mathbf{1}_{ij}^{obj}[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord}\Sigma_{i=1}^{S^2}\Sigma_{j=1}^{B}\mathbf{1}_{ij}^{obj}[(w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2] \tag{7}$$

where λ_{coord} is the weight of the coordinate error, S^2 is the number of grids in the input image, and B is the number of bounding boxes generated by each grid. Referring to the original parameters in the YOLO-V3 model, $\lambda_{coord} = 5$, $S = 7$, and $B = 9$ were selected in this study. $\mathbf{1}_{ij}^{obj} = 1$ denotes that the object falls into the *jth* bounding box in grid *i*, otherwise $\mathbf{1}_{ij}^{obj} = 0$. $(\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i)$ are values of the center coordinate, height, and width of the predicted bounding box. (x_i, y_i, w_i, h_i) are true values.

The IoU error $Error_{iou}$ is defined as follows:

$$Error_{iou} = \Sigma_{i=1}^{S^2}\Sigma_{j=1}^{B}\mathbf{1}_{ij}^{obj}(C_i - \hat{C}_i)^2 + \lambda_{noobj}\Sigma_{i=1}^{S^2}\Sigma_{j=1}^{B}\mathbf{1}_{ij}^{obj}(C_i - \hat{C}_i)^2 \tag{8}$$

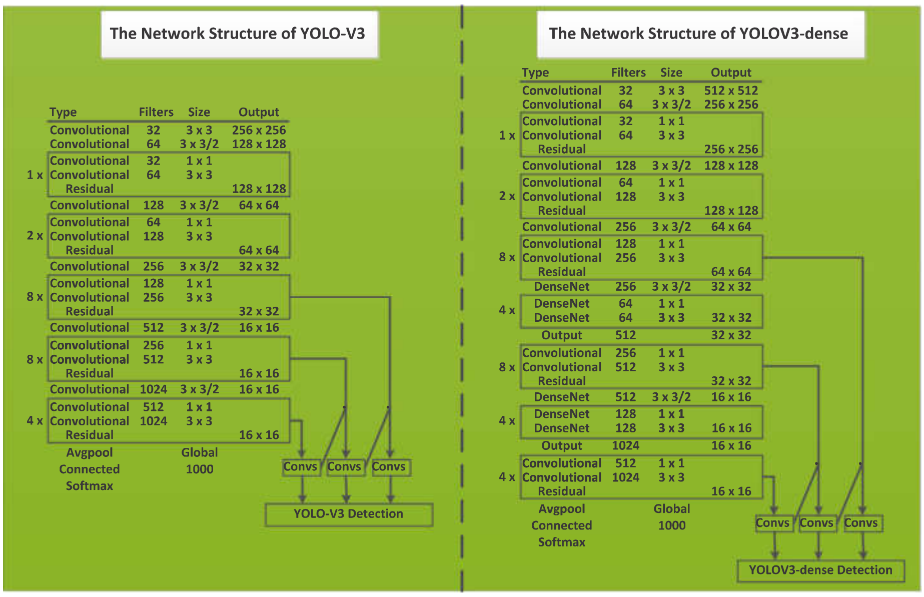
where the parameter λ_{noobj} is the weight of the IoU error. Referring to the original parameters of the YOLO-V3 model, $\lambda_{noobj} = 0.5$ was selected in this paper. $\hat{C}_i$ is the predicted confidence, and C_i is the true

Table 2
Initialization parameters of YOLOV3-dense network.

Size of input images	Batch	Momentum	Initial learning rate	Decay	Training steps
512 × 512	8	0.9	0.001	0.0005	70,000

Table 3
Confusion matrix for the classification results.

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



The Network Structure of YOLO-V3

	Type	Filters	Size	Output
	Convolutional	32	3 x 3	256 x 256
	Convolutional	64	3 x 3/2	128 x 128
1 x	Convolutional	32	1 x 1	
	Convolutional	64	3 x 3	
	Residual			128 x 128
	Convolutional	128	3 x 3/2	64 x 64
2 x	Convolutional	64	1 x 1	
	Convolutional	128	3 x 3	
	Residual			64 x 64
	Convolutional	256	3 x 3/2	32 x 32
8 x	Convolutional	128	1 x 1	
	Convolutional	256	3 x 3	
	Residual			32 x 32
	Convolutional	512	3 x 3/2	16 x 16
8 x	Convolutional	256	1 x 1	
	Convolutional	512	3 x 3	
	Residual			16 x 16
	Convolutional	1024	3 x 3/2	16 x 16
4 x	Convolutional	512	1 x 1	
	Convolutional	1024	3 x 3	
	Residual			16 x 16
	Avgpool		Global	
	Connected		1000	
	Softmax			

The Network Structure of YOLOV3-dense

	Type	Filters	Size	Output
	Convolutional	32	3 x 3	512 x 512
	Convolutional	64	3 x 3/2	256 x 256
1 x	Convolutional	32	1 x 1	
	Convolutional	64	3 x 3	
	Residual			256 x 256
	Convolutional	128	3 x 3/2	128 x 128
2 x	Convolutional	64	1 x 1	
	Convolutional	128	3 x 3	
	Residual			128 x 128
	Convolutional	256	3 x 3/2	64 x 64
8 x	Convolutional	128	1 x 1	
	Convolutional	256	3 x 3	
	Residual			64 x 64
	DenseNet	256	3 x 3/2	32 x 32
4 x	DenseNet	64	1 x 1	
	DenseNet	64	3 x 3	32 x 32
	Output	512		32 x 32
8 x	Convolutional	256	1 x 1	
	Convolutional	512	3 x 3	
	Residual			32 x 32
	DenseNet	512	3 x 3/2	16 x 16
4 x	DenseNet	128	1 x 1	
	DenseNet	128	3 x 3	16 x 16
	Output	1024		16 x 16
4 x	Convolutional	512	1 x 1	
	Convolutional	1024	3 x 3	
	Residual			16 x 16
	Avgpool		Global	
	Connected		1000	
	Softmax			

Fig. 4. Network parameters of YOLO-V3 and YOLOV3-dense.

confidence.

The classification error $Error_{cls}$ is defined as follows:

$$Error_{cls} = \Sigma_{i=1}^{S^2}\Sigma_{j=1}^{B}\mathbf{1}_{ij}^{obj}\Sigma_{c\in classes}(p_i(c) - \hat{p}_i(c))^2 \tag{9}$$

where c refers to the class to which the detected target belongs. $p_i(c)$ refers to the true probability that the object belonging to class c is in grid i. $\hat{p}_i(c)$ is the predicted value. The $Error_{cls}$ for grid i is the sum of classification errors for all the objects in the grid.

C. IoU

IoU is a standard for defining the detection accuracy of target objects. IoU evaluates the performance of the model by calculating the overlap ratio between the predicted bounding box and the true bounding box as follows:

$$IoU = \frac{S_{overlap}}{S_{union}} \tag{10}$$

where $S_{overlap}$ is the area of intersection of the predicted bounding box and the true bounding box. S_{union} is the area of the union of the two bounding boxes.

Detection Time

The average detection times for several deep learning models were compared in this paper, and the real-time performance of these models was analysed.

4.1. Influence of data category

To compare the effect of data category on the detection results, the YOLOV3-dense neural network was used to train images of young, expanding, and ripe apples, respectively. The images of apples in these three periods were also combined and used to train the model. The P-R curves of the models after training are shown in Fig. 5. The F1 scores of the corresponding models are shown in Table 4.

In order to observe the boundary boxes better, 1, 2, and 3 are used to label young, expanding, and ripe apples, respectively. The results from the corresponding models are shown in Fig. 6.

Based on the above detection results, the F1 score of the model trained using apple images during one growth stage is higher than that of the model trained by combining the images together. The model trained using apple images during one growth stage can better detect some apples under serious occlusion and overlap. This indicates that the number of input classes will affect the detection ability of the model. Because the young apple fruits are relatively small in volume and densely overlap each other, the detection results for young apples are worse than those for expanding and ripe apples. The model shows the best detection performance for ripe apples due to the more obvious colour characteristics, larger individual volume, and less overlap.

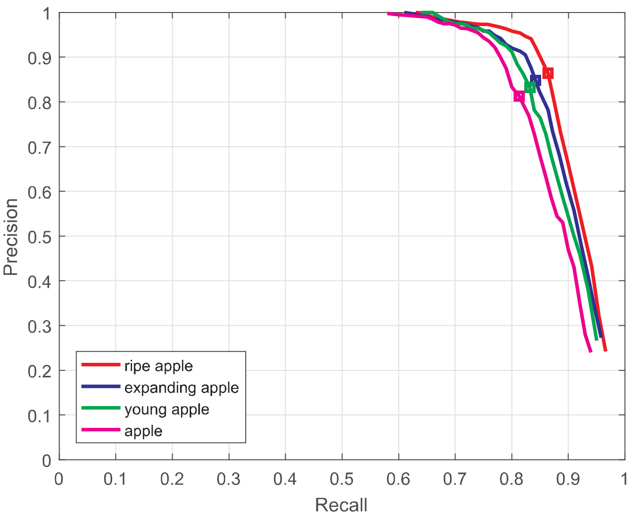


Fig. 5. P-R curves of apple detection models in several categories.

Table 4
F1 scores of apple detection models in several categories.

Class	F1 score
Young apple	0.832
Expanding apple	0.841
Ripe apple	0.864
All	0.817

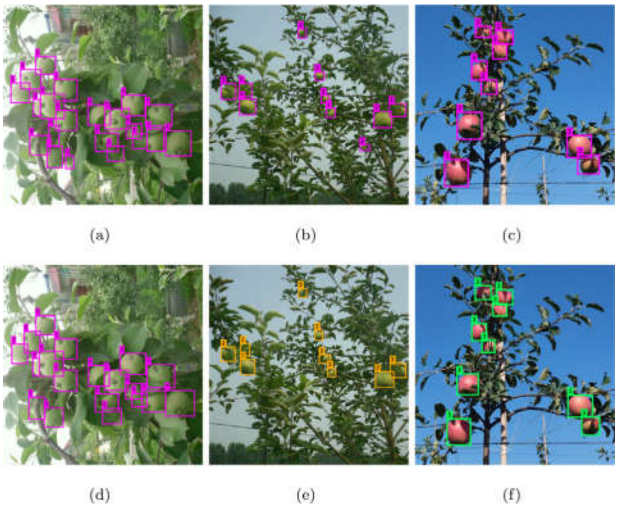


Fig. 6. Detected apples in several growth stages: (a) images of young apples, (b) images of expanding apples, (c) images of ripe apples, (d-f) images of apples in three growth stages detected by the same model.

4.2. Comparison of different algorithms

In order to verify the performance of the model proposed in this paper, images of apple in the three growth stages are used as the training set. The proposed model is compared with YOLO-V2, YOLO-V3, and Faster R-CNN with VGG16 net, which is the state-of-art fruit detection model in order to illustrate the superiority of the YOLOV3-dense model proposed in this paper.

The loss of YOLO-V2, YOLO-V3 and YOLOV3-dense during training is shown in Fig. 7.

The P-R curves for several models during testing are shown in Fig. 8. The F1 scores, IoU, and average detection time of the models are shown

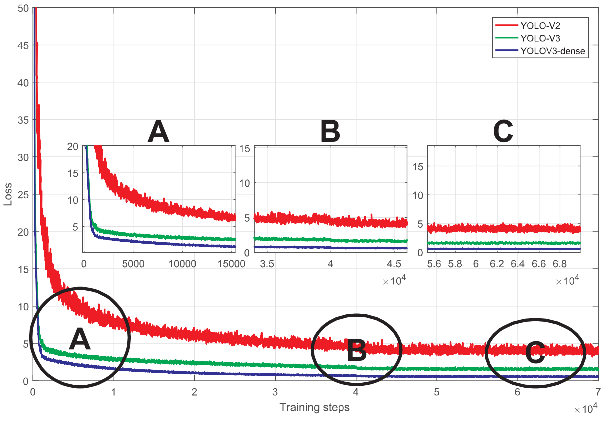


Fig. 7. Loss curves of the three YOLO models.

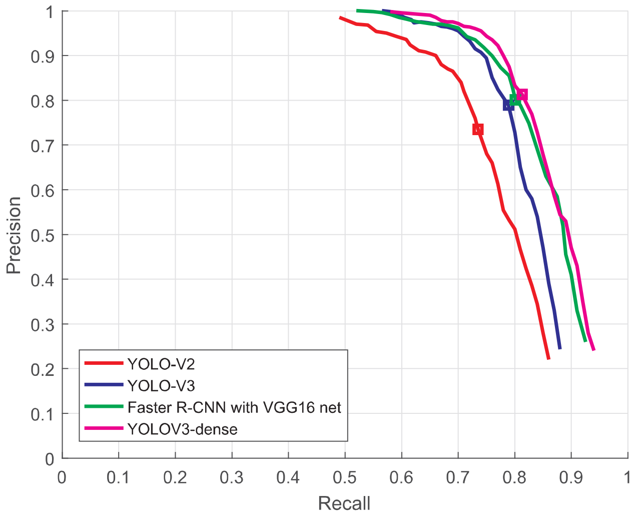


Fig. 8. P-R curves for the detection models.

Table 5
F1 Scores, IoU and average detection time for several models.

Models	YOLO-V2	YOLO-V3	Faster R-CNN with VGG16 net	YOLOV3-dense
F1 score	0.738	0.793	0.801	0.817
IoU	0.805	0.869	0.873	0.896
Average time (s)	0.273	0.296	2.42	0.304

in Table 5.

The detection results for models in different growth stages are shown in Fig. 9 and Table 6.

Based on the above results, one can see that YOLO-V3 has faster convergence speed and better convergence results than YOLO-V2 during training. The final loss in YOLO-V2 is around 3.95, while the loss in YOLO-V3 is around 1.53. The loss in YOLOV3-dense is about 0.54, which is about 0.99 lower than the original YOLO-V3 model. This shows that the performance of the proposed model is significantly improved. The loss curve for YOLO-V3 began to saturate after 3000 training steps. However, the loss for YOLOV3-dense continues to converge up to 45,000 steps, after which it no longer decreases. In terms of detection performance, the proposed YOLO-V3 dense model is superior to the Faster R-CNN with VGG16 net, YOLO-V3, and YOLO-V2 models. The F1 score of YOLOV3-dense is 0.817, which is higher than the other three models. This indicates that the comprehensive recall performance and precision of the YOLOV3-dense model is better than that of the other three models. The IoU value of YOLOV3-dense is 0.896, which is higher than that of the other three models. This result shows that the accuracy of YOLOV3-dense in detecting bounding boxes is higher than that of the other three models. The average detection time of YOLOV3-dense is 2.116*s* less than Faster R-CNN with VGG16 net and is basically the same as the YOLO-V3 model. It can provide real-time detection of apples in high resolution images. During detection, the accuracy and confidence provided by the YOLOV3-dense model is significantly higher than the other three models, reflecting the superiority of the YOLOV3-dense detection model.

4.3. Influence of the quantity of experimental data

In this section, the impact of the size of the image dataset on the YOLOV3-dense model is analysed. 10, 50, 200, 400, 800, 1200, and 1600 apple images were randomly selected from each of the three growth stages to form training sets of 30, 150, 600, 1200, 2400, 3600, and 4800 images. The P-R curves and F1 scores for the models corresponding to training sets of different sizes are shown in Fig. 10 and Table 7.

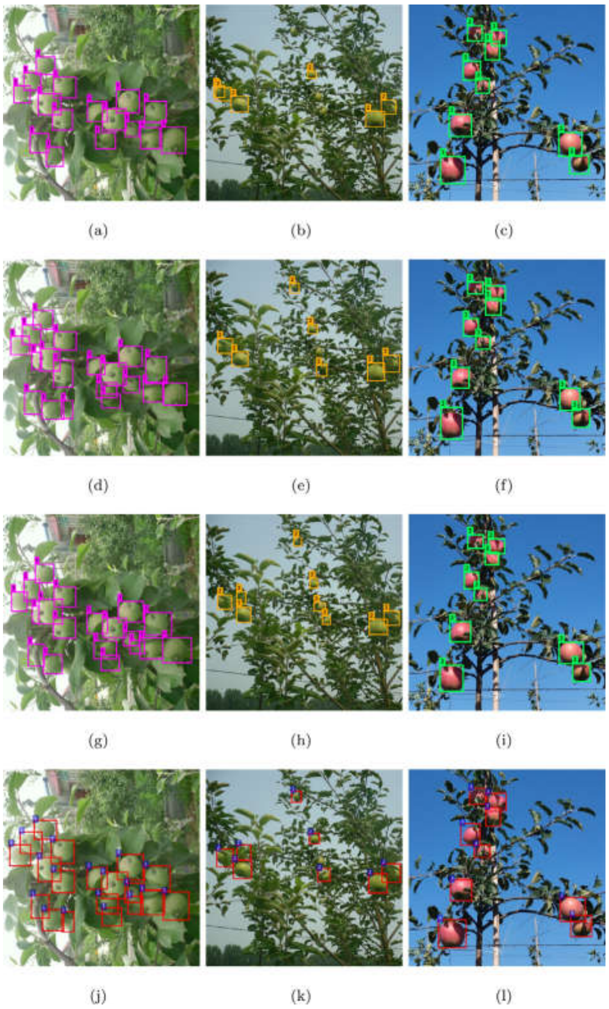


Fig. 9. Detection results of the four models: (a–c) YOLO-V2, (d–f) YOLO-V3, (g–i) YOLOV3-dense, (j–l) Faster R-CNN with VGG16 net.

From these experiments, one can draw the conclusion that the performance of the YOLOV3-dense model improves as the size of the training set increases. If the training set contains fewer than 1500 images, the performance enhances rapidly as the training set grows. When the size of training set exceeds 1500, the enhancement speed gradually reduces as the amount of images increases. When the amount of images exceeds 3000, the size of the training set does not have a further significant influence on the performance of the model.

4.4. Influence of data augmentation methods

Color, brightness, rotation transformation and blur processing were used to augment the images. In order to verify the influence of the four transformation methods on the training model, the control variable method was used to remove one data augmentation method each time and obtain the IoU value and F1 score. The influence of the multi-angle viewing method during image acquisition on the detection performance was also considered. After data augmentation, the number of images obtained by multi-angle viewing method increases to 880. The results are shown in Table 8.

Based on the experimental results, one can see that multi-angle image acquisition is of great help to simulate multi-angle viewing during detection. Removing this method, the F1 score of the detection model decreases by 0.033 and the IoU decreases by 0.058, indicating that the performance of the model decreases significantly. Therefore, the

Table 6
Detection results of different models for apples in different growth stages in Fig. 9.

Figure	Number of apples detected	Confidence
a	16	86%,85%,85%,85%,82%,82%,79%,70%, 69%,69%,66%,64%,64%,58%,52%,50%
b	5	84%,77%,73%,63%,50%
c	9	87%,87%,84%,84%,83%,83%,71%,70%, 69%
d	19	100%,100%,100%,100%,100%,100%,100%,99%, 99%,99%,99%,98%,98%,95%,92%,84%,83%,75%,54%
e	7	100%,100%,99%,90%,73%,81%,60%
f	9	100%,100%,100%,100%,100%,100%,100%,99%, 62%
g	19	100%,100%,100%,100%,100%,100%,100%,100%,100%,100%,100%,100%,98%,92%,82%,79%, 77%,66%,56%
h	9	100%,100%,99%,99%,99%,96%,91%,79%,70%
i	9	100%,100%,100%,100%,100%,100%,99%,99%, 75%
j	19	100%,100%,100%,100%,100%,100%,99%,99%, 95%,92%,92%,88%,83%,82%,82%,75%,63%,60%,52%
k	8	100%,97%,97%,95%,91%,72%,65%,62%
l	9	100%,100%,100%,100%,100%,98%,95%,90%, 75%

multi-angle image viewing method is conducive to improving the performance of the model.

The colour balance transformation is very helpful for improving detection. Removing the colour balance transformation will reduce the detection accuracy.

The brightness transformation is beneficial for the model to adapt to the illumination situations throughout the day. The detection results of the model trained by removing the brightness transformation is worse than that of the model trained with the complete dataset.

The rotation transformation has limited effect on the training model, and the performance of the training model after removing the rotation transformation is slightly lower than that of the complete dataset.

Blur processing is quite favourable for improving the robustness of the model. Compared with the dataset without blur processing, the model trained with the complete dataset has much greater detection accuracy.

4.5. Detection under occlusion and overlapping apples conditions

In orchards, partial occlusion of branches and leaves commonly occur, as well as overlap between apples. This would have a certain influence on apple detection. In this section, the IoU values and F1 scores from the YOLOV3-dense model with occluded and overlapping apples are analysed. The results are shown in Table 9 and Fig. 11.

Based on the above experiments, one can see that occluded and overlapped apples cause inaccurate detection. However, the model can be used to detect most of the occluded and overlapped apples, which illustrates the practical significance of the model proposed in this paper.

Table 7
F1 scores of models trained with different numbers of images.

Number of images	30	150	600	1200	2400	3600	4800
F1 score	0.476	0.642	0.714	0.755	0.786	0.808	0.817

Table 8
F1 scores and IoU values for models trained using the control variable method.

Data augmentation method	F1 score	IoU
Dataset after augmentation	0.817	0.896
Remove multi-angle viewing method	0.784	0.838
Remove color balance transformation	0.787	0.842
Remove brightness transformation	0.795	0.854
Remove rotation transformation	0.808	0.882
Remove blur processing	0.766	0.829

Table 9
IoU values and F1 scores in the YOLOV3-dense model with occluded and overlapping apples.

Growth stage	IoU	F1 score
Young apple	0.874	0.783
Expanding apple	0.882	0.791
Ripe apple	0.889	0.809

4.6. Detection in an environment without apples

In a real setting, the camera can also capture images that do not contain apples. In this paper, 50 images that do not include apples were collected to test the performance of the detection model in a real setting. Among them, 10 images contain only the sky, 10 images contain only the ground, 10 images contain only trees without apples, and 20 images contain these three possible backgrounds. These 50 images were tested using the YOLOV3-dense model. The detection results showed that no apples were detected in these 50 images.

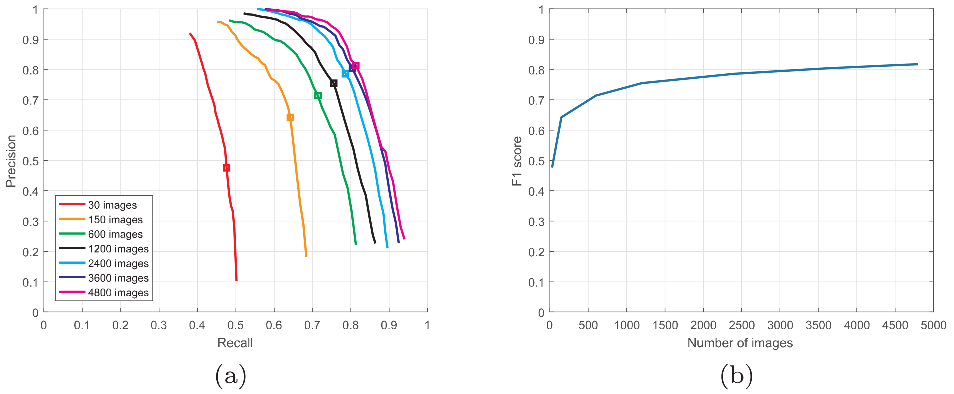


Fig. 10. P-R curves and F1 scores of the model trained with different numbers of images.

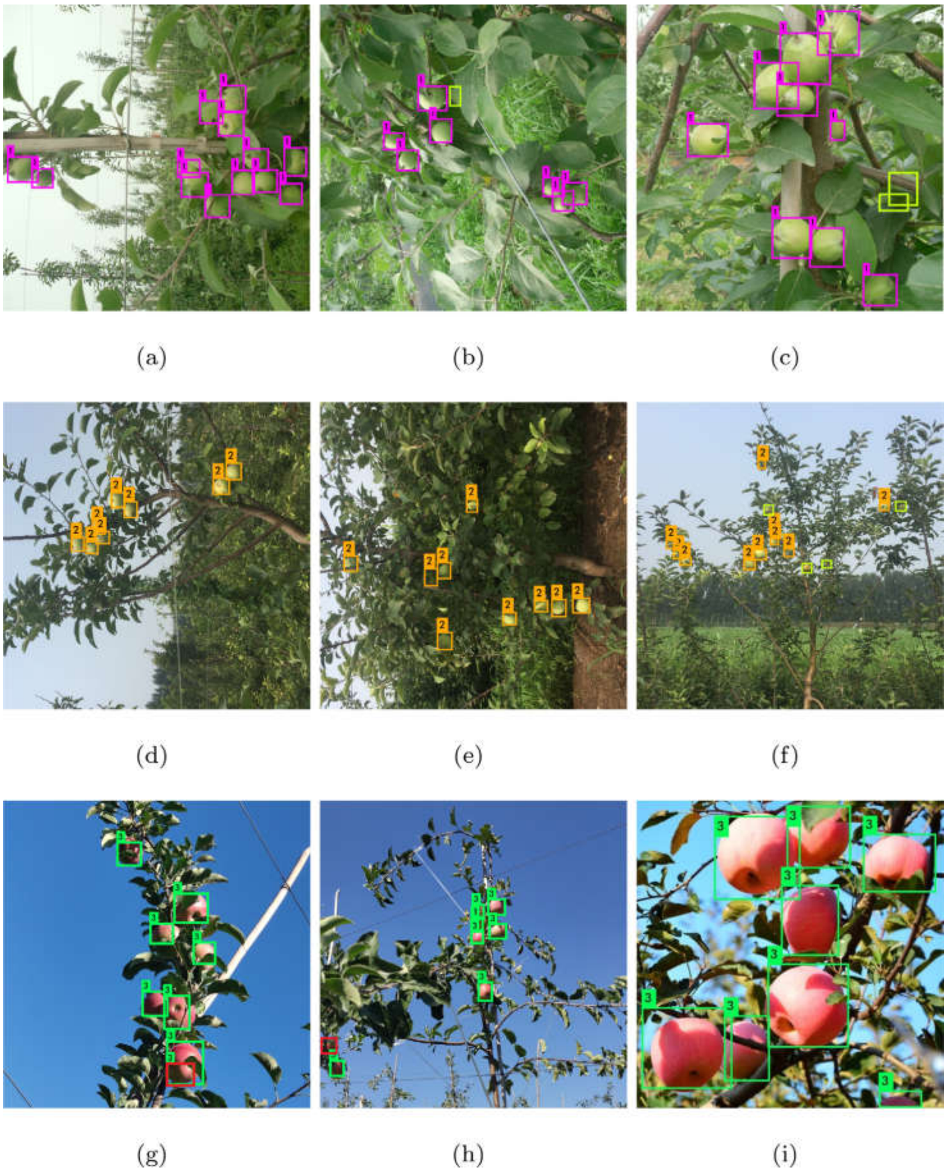


Fig. 11. Detection results for occluded and overlapped apples: (a–c) young apples, where apples in yellow-green boxes are missed and mistaken apples; (d–f) expanding apples, were apples in yellow-green boxes are missed and mistaken apples; (g–i) expanding apples, where apples in red boxes are missed and mistaken apples. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

From the experimental results, one can see that the YOLOV3-dense detection model can properly identify the backgrounds. In fact, during training, the areas that were not framed during manual annotation will be labelled as the background by default. Therefore, during training, the actual input contains 4 categories, including the background and the apples in the three growth stages. The results of this experiment further demonstrate that the YOLOV3-dense detection model can provide high classification accuracy.

5. Conclusions

In this study, the state-of-art YOLO-V3 detection model was improved by incorporating the DenseNet method for detecting apples in the main growth stages in orchards. This model can be used to detect young apples, expanding apples, and ripe apples. The YOLOV3-dense model proposed in this paper uses DenseNet to optimize the feature layers with low resolution in the YOLO-V3 model by enhancing feature propagation, promoting feature reuse, and improving network performance. The experimental results show that the YOLOV3-dense model proposed in this paper has better performance compared to the YOLO-V3 model and is superior to the Faster R-CNN with VGG16 net, which is the state-of-art fruit detection model. The YOLOV3-dense model can also be used to detect occluded and overlapping apples in real-time.

Future work will focus on applying existing models to detecting apples in videos, yield estimation, and other practical tasks. The environmental characteristics and characteristics of apples in different growth stages which were not involved in this paper will be analysed. In addition, data augmentation methods and the detection model will be optimized to further improve the detection accuracy.

Acknowledgements

This project was supported by National Key Research and Development Program of China (Grant No. 2017YFD0701401).

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.compag.2019.01.012.

References

Arribas, J.I., Sánchez-Ferrero, G.V., Ruiz-Ruiz, G., Gómez-Gil, J., 2011. Leaf classification in sunflower crops by computer vision and neural networks. Comput. Electron. Agric. 78, 9–18.

Bargoti, S., Underwood, J., 2016. Deep fruit detection in orchards. Aust. Centre Field Robotics 1–8.

Chen, S.W., Skandan, S.S., Dcunha, S., Das, J., Okon, E., Qu, C., Taylor, C.J., Kumar, V., 2017. Counting apples and oranges with deep learning: a data driven approach. IEEE Robotics Automation Lett. 2, 781–788.

Dias, P.A., Tabb, A., Medeiros, H., 2018. Apple flower detection using deep convolutional networks. Comput. Ind. 99, 17–28.

Dyrmann, M., Jørgensen, R.N., Midtiby, H.S., 2017. RoboWeedSupport - detection of weed locations in leaf occluded cereal crops using a fully convolutional neural network. Adv. Anim. Biosci.: Precision Agric. 8, 842–847.

Hamuda, E., Ginley, B.M., Glavin, M., Jones, E., 2018. Improved image processing-based crop detection using kalman filtering and the hungarian algorithm. Comput. Electron. Agric. 148, 37–44.

Huang, G., Liu, Z., Laurens, V.D.M., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: IEEE conference on Computer Vision and Pattern Recognition, pp. 2261–2269.

Inkyu, S., Ge, Z., Feras, D., Ben, U., Tristan, P., Chris, M.C., 2016. DeepFruits: a fruit detection system using deep neural networks. Sensors 16, 1222.

Kamilaris, A., Prenafeta-Boldú, F.X., 2018. Deep learning in agriculture: a survey. Comput. Electron. Agric. 147, 70–90.

Lam, E.Y., 2005. Combining gray world and retinex theory for automatic white balance in digital photography. In: International Symposium on Consumer Electronics, pp. 134–139.

Lee, S.H., Chan, C.S., Mayo, S.J., Remagnino, P., 2017. How deep learning extracts and learns leaf features for plant classification. Pattern Recogn. 71, 1–13.

Linker, R., Cohen, O., Naor, A., 2012. Determination of the number of green apples in RGB images recorded in orchards. Comput. Electron. Agric. 81, 45–57.

Lu, J., Sang, N., 2015. Detecting citrus fruits and occlusion recovery under natural illumination conditions. Comput. Electron. Agric. 110, 121–130.

Rahnemoonfar, M., Sheppard, C., 2017. Deep count: fruit counting based on deep simulated learning. Sensors 17, 905.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: unified, real-time object detection. In: IEEE conference on Computer Vision and Pattern Recognition, pp. 779–788.

Redmon, J., Farhadi, A., 2017. YOLO9000: Better, faster, stronger. In: IEEE conference on Computer Vision and Pattern Recognition, pp. 6517–6525.

Redmon, J., Farhadi, A., 2018. YOLOv3: An incremental improvement. In: IEEE conference on Computer Vision and Pattern Recognition, arXiv:1804.0276.

Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Machine Intelligence 39, 1137–1149.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. Comput. Sci arXiv:1409.1556.

Tang, J.L., Wang, D., Zhang, Z.G., He, L.J., Xin, J., Xu, Y., 2017. Weed identification based on K-means feature learning combined with convolutional neural network. Comput. Electron. Agric. 135, 63–70.

Tyagi, A.C., 2016. Towards a second green revolution. Irrigation Drainage 65, 388–389.

Wang, Q., Nuske, S., Bergerman, M., Singh, S., 2013. Automated crop yield estimation for apple orchards. Exp. Robotics 88, 745–758.

Yamamoto, K., Guo, W., Yoshioka, Y., Ninomiya, S., 2014. On plant detection of intact tomato fruits using image analysis and machine learning methods. Sensors 14, 12191–12206.

Zhang, Y., Phillips, P., Wang, S., Ji, G., Yang, J., Wu, J., 2016. Fruit classification by biogeography-based optimization and feedforward neural network. Expert Syst. J. Knowledge Eng. 33, 239–253.

Zhang, Y.D., Dong, Z., Chen, X., Jia, W., Du, S., Muhammad, K., Wang, S.H., 2017. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. Multimedia Tools Appl. 1–20.

Zhao, Y., Gong, L., Huang, Y., Liu, C., 2016. A review of key techniques of vision-based control for harvesting robot. Comput. Electron. Agric. 127, 311–323.