Computers and Electronics in Agriculture 176 (2020) 105634

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Original papers

Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN

Fangfang Gao[a], Longsheng Fu[a,b,c,d,*], Xin Zhang[d], Yaqoob Majeed[d], Rui Li[a], Manoj Karkee[d], Qin Zhang[d]

[a] College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling, Shaanxi 712100, China
[b] Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture and Rural Affairs, Yangling, Shaanxi 712100, China
[c] Shaanxi Key Laboratory of Agricultural Information Perception and Intelligent Service, Yangling, Shaanxi 712100, China
[d] Centre for Precision and Automated Agricultural Systems, Washington State University, Prosser, WA 99350, USA

ARTICLE INFO

Keywords:
Branch/wire-occluded fruit
Deep learning
Data augmentation
Multi-class detection
Robotic harvesting

ABSTRACT

Deep learning achieved high success of fruit-on-plant detection such as on apple. Most of studies on apple detection identified all target fruits as one class regardless of fruit condition and other canopy objects. However, some detected fruits were physically occluded by branches or trellis wires that could diminish the effectiveness of fruit picking and even damage the end-effector, especially when high-vigor rootstock apple cultivar was used. A multi-class apple detection method in dense-foliage fruiting-wall trees was thus proposed based on Faster Region-Convolutional Neural Network. It detected apples in different conditions such as non-occluded, leaf-occluded, branch/wire-occluded, and fruit-occluded fruit. A total of 800 images were acquired and then augmented to 12,800 images. Average precision of non-occluded, leaf-occluded, branch/wire-occluded, and fruit-occluded fruit were 0.909, 0.899, 0.858, and 0.848, respectively. Overall, the mean average precision of the four classes was 0.879, and an average of 0.241 s was needed to process an image. The results indicated that all the apples in different classes could be effectively detected, which can help the robot to decide the picking strategy (e.g., picking order and path planning) as well as to avoid the potential damage by the branches and trellis wires.

1. Introduction

Apple is one of the most valuable fruit crops around the world. In 2017, farmers produced around 83 million tons of apples with the planting area of approximately 5 million hectares in the world (Wang et al., 2019). However, with that large of production, handpicking with ladder-and-bucket and harvest platforms is currently still the main harvesting method for the fresh market apples, which is labor-intensive and costly. Moreover, seasonal task of picking creates a great challenge for growers to find sufficient labor for timely harvesting (Zhang et al., 2018a,b; Feng et al., 2019; Zhao et al., 2011). To reduce dependence on human labor, it is of significant importance to replace manual picking with mechanical/robotic harvesting.

Automatic harvesting has become an essential requirement due to reduced labor availability and increased labor costs. The main advantage of automatic harvesting is its ability to facilitate selective harvesting and potential to reduce the dependence on labor force (Zhang et al., 2016). Most fruit trees, especially the apple trees, are now in a time of transformation from traditional to modern orchards (Majeed et al., 2020). In the past, a major challenge for automation development in apple production was the complexity of apple tree architectures. In recent years, planting of new apple orchards are adopting fruiting-wall architectures, more generically called SNAP (i.e., simple, narrow, accessible, and productive) systems. In this architecture, trees are tied by tapes to wires (Fig. 1) and planted close together with interplant spacing generally ranging from 0.3 to 1.5 m, while inter-row spacing varies typically from 2.5 to 4.0 m (Zhang et al., 2018a,b). Compared to conventional tree architecture, the SNAP systems provide a simpler tree structure and easier access to fruits, leaves, and branches for various automated field operations such as harvesting, pruning, and spraying.

Fruit detection is the primary key technology for automatic harvesting and has been extensively studied using traditional image processing technology (Fu et al., 2019; Tang et al., 2020). Liu et al. (2019) obtained a 90.6% detection rate using the support vector machine (SVM) classifier with Gaussian kernel function to detect apples. Feng et al. (2019) developed an algorithm for detecting apples with a detection rate of 82.8% based on the pseudo-color and texture

* Corresponding author.
E-mail address: fulsh@nwafu.edu.cn (L. Fu).

https://doi.org/10.1016/j.compag.2020.105634
Received 24 May 2020; Received in revised form 24 June 2020; Accepted 10 July 2020
Available online 18 July 2020
0168-1699/

(a) (b)

Fig. 1. (a) The tree architecture tied by wires in a modern SNAP orchard; (b) The specific details of a tree tied by tapes to wire.

information from multi-spectral dynamic images. Gongal et al. (2015) reviewed the development of machine vision systems for fruit detection for robotic harvesting, and concluded that the accuracy on fruit detection was about 0.70–0.92 for apples. These methods can detect the object regions, but produce errors for densely distributed and heavily overlapped objects (Lv et al., 2016; Lin et al., 2020; Nguyen et al., 2016). The different features of fruits and complex background in the orchard constantly make the traditional object detection techniques more challenging to achieve desired accuracy.

With the rapid development of machine learning, deep learning technology has been widely used in fruit detection from 2016 and achieved outstanding results (Wang and He, 2019). Koirala et al. (2019a) reviewed the fruit detection using deep learning and concluded that most studies achieved higher detection rates of more than 84%. Among all available techniques and networks, the Faster Region-based Convolutional Neural Network (Faster R-CNN) is one of the most commonly applied techniques for small object recognition and, therefore, for fruit detection in field conditions (Sa et al., 2016; Zhang et al., 2018a,b; Gené-Mola et al., 2019; Häni et al., 2020; Liu et al., 2020; Wan and Goudos, 2020). Häni et al. (2020) employed Faster R-CNN for detecting apple on untrained tree and achieved a success rate of 90.8% on 103 images under natural illumination. Gené-Mola et al. (2019) used Faster R-CNN to detect tall spindle 'Fuji' apples images that captured at night with artificial lighting and obtained an average precision (*AP*) of 0.948 on 967 images with 12,839 fruits. Liu et al. (2019) used Faster R-CNN based VGG16 (Visual Geometry Group with 16 layers) to detect apple images with few fruits under natural illumiantions and achieved a F1-score of 90.57%. Wan and Goudos (2020) also applied VGG16 to detect apple images with few fruits under natural illumiantions and obtained an *AP* of 0.925 on 820 images. Those studies obtained acceptable results on fruit detection using Faster R-CNN.

The deep learning method achieved a high detection rate and fast detection speed. Most of the studies on apple detection identified all the target fruits as one class, including fruits that were partly visible but occluded by branches or trellis wires, as shown in Fig. 2. Kang and Chen (2020) proposed a deep learning model LedNet for 'Fuji' apple detection and reached 100% accuracy with 0.028 s to process an image, including fruits occluded by branches (Fig. 2a). Tian et al. (2019) improved another deep learning model YOLOv3 to detect apples at different growth stages, among which apples occluded by multiple branches were also detected successfully (Fig. 2b). However, the apples occluded by branches or wires are difficult to be picked by current linear apple-picking robots (Silwal et al., 2017; Zhang et al., 2016). The robotic end-effector may potentially be damaged if the robot forcibly picks the apples that occluded by branches or wires, where unpredictable economic losses can be caused. To avoid that, apples occluded by branches or wires should not be picked by robots. Therefore, the apples should be detected as multiple classes. In that case, the robot can decide the picking strategy (e.g., picking order and path planning) as well as to avoid the potential damage by the branches and trellis wires according to the detected classes of apples for robotic picking.

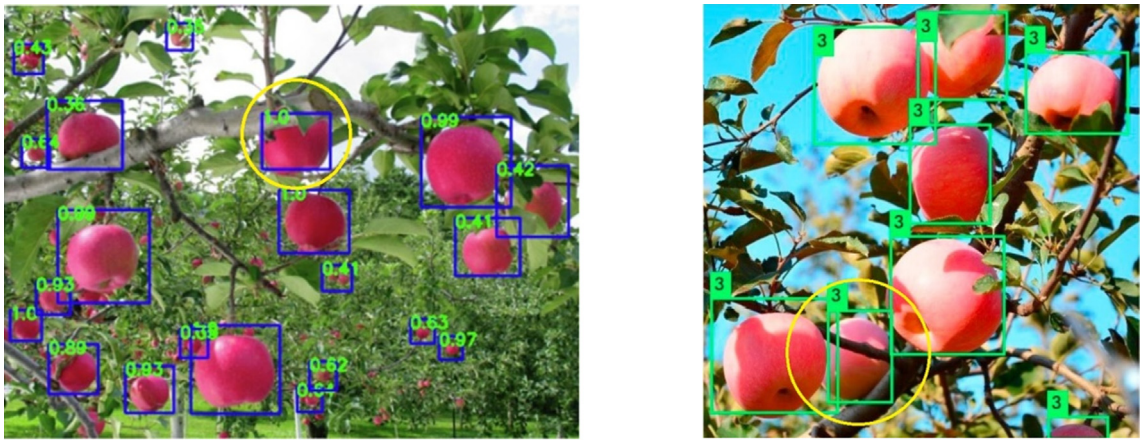
In this study, a deep learning based multi-class fruit detection method is proposed to identify the apples as multiple classes. VGG16, a widely used model of the Faster R-CNN, was adapted and implemented for the purpose. The rest of the paper is organized as follows. In Section 2, the materials and methods are described in terms of pre-processing the image dataset and detection algorithm. In Section 3, the results are presented and discussed. Lastly, in Section 4, the conclusions and prospects of this paper are described.

2. Materials and methods

This study focused on detecting apples in vertical fruiting-wall tree architecture, which is one of the SNAP systems commonly planted in the orchards of Washington state (WA), USA. A total of 800 RGB (Red, Green, and Blue) images were acquired in a commercial apple orchard near Prosser, WA. All images were annotated manually with rectangular annotations and generated corresponding 'xml' annotation files were saved. The annotated images were augmented using geometric transformation and image enhancement. All images were used as the inputs to Faster R-CNN based VGG16 model for training and testing. Performance of VGG16 were evaluated and compared with Zeiler and Fergus Network (ZFNet) by the evaluation measures of precision (*P*), recall (*R*), *AP*, and mean average precision (*mAP*). Detailed explanations of the method are provided in the following sections.

2.1. Image acquisition

Image data were collected during two harvesting seasons (2017 and 2018) in a commercial orchard, as shown in Fig. 3a. For acquiring fruit-on-plant images, a portable imaging platform with a Microsoft Kinect V2 sensor (Microsoft Inc., Redmond, WA, USA), a camera clamping device, artificial LED (light-emitting diode) lightings (Trilliant 36 Light Emitting Diode Grote, Madison, IN, USA) and a support frame was created, as shown in Fig. 3b. The mounting height of the camera was adjusted by the clamping device. The Kinect V2 sensor was mounted at the center of four LED lights on the platform to maintain a vertical distance of ~1.7 m from the ground and kept the range of 0.5 m to the target fruiting-wall to capture apple images on the 3rd (~1.5 m height) and 4th (~2.0 m height) layers of the tree canopies. The four LED lights were installed in the platform to create a uniform lighting environment



(a) (b)

Fig. 2. Apples occluded by branches (yellow circles) were detected in other studies using deep learning methods. (a) Kang and Chen (2020) proposed LedNet for Fuji apple detection and reached 100% accuracy, including fruits occluded by branches; (b) Tian et al. (2019) improved YOLOv3 model to detect apple at different growth stages, among which apple occluded by branches was also detected successfully. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

for nighttime imaging, as shown in Fig. 3c. Lighting was not controlled specifically in this study because of the more variation the images have for deep learning networks, the better generalization capabilities can be achieved with higher accuracy and robustness.

Change of sun angle naturally caused the variation in the images will help to develop more robust detection system. During image acquisition, camera's viewing direction was either set parallel to sunlight direction in order to simulate front-lighting, and the rendering was shown in Fig. 4a; or set antiparallel to simulate backlighting, and the rendering was shown in Fig. 4b. In total, 800 apple images with 1920 × 1080 pixels were collected under different lighting conditions to improve the robustness of the network.

2.2. Fruit classes

As mentioned earlier, in the orchard, a large amount of the apples is occluded, which could cause difficulties for robotic picking. Therefore, apples were categorized into four classes according to the occlusion condition of apples in this study. The first class (i.e., leaf-occluded fruit) refers to fruits that are occluded by leaves (Fig. 5a). The second class (i.e., branch/wire-occluded fruit), as shown in Fig. 5b and c, refers to the fruits are occluded by branches or wires, whether or not it is occluded by leaves or other fruits. Fruits of this class cannot be picked by robots. The third class is non-occluded fruit (Fig. 5d), where apples were completely independent and separated from leaves, branches, wires, and other apples. Fruits of this class can be picked directly by robot with priority. The fourth class refers to the fruits that overlap with each other (i.e., fruit-occluded fruit, Fig. 5e), where the outside one should be picked first. All fruits were detected as above mentioned four classes, by which the robot can make appropriate decisions.

The apples were manually annotated into the four classes with rectangular annotations as the ground-truth images, as shown in Fig. 6. A label was added onto each rectangular annotation to indicate the class of apple, where C01 refers to the leaf-occluded fruit, C02 refers to the branch/wire-occluded fruit, C03 refers to the non-occluded fruit, and C04 refers to the fruit-occluded fruit. The image labeling software used in this study was LabelImg and anootations were saved in 'xml' format.

(a) (b) (c)

Fig. 3. (a) Experimental orchard used in this work; (b) Self-built platform for image acquisition; and (c) Distribution of the light sources.

Fig. 4. (a) Image acquired in the front-lighting; (b) Image acquired in the backlighting.

Fig. 5. Apples were categorized into four classes according to the occlusion condition. (a) Leaf-occluded fruit, which can be picked by robot; (b) and (c) Branch/wire-occluded fruit, which cannot be linearly picked by robot; (d) Non-occluded fruit, which has the first order to be picked by robot; (e) Fruit-occluded fruit, where the outer fruit before it should be picked first and then the inner fruit.

2.3. Data augmentation

Small number of training images may lead to overfitting or non-convergence of the deep learning algorithm. While increasing the amount of training images using data augmentation could be an effective way to solve this problem (Huang et al., 2020). Data augmentation, including geometric transformation and data enhancement, was implemented using the software Matlab 2018b with the Image Processing Toolbox in this study. The specific augmented methods were described as follows.

Among data augmentaion techniques, image mirroring and rotations were performed as geometric transformation. Image mirroring (horizontal and vertical mirror) was implemented using the Matlab function 'flip' (Luus et al., 2015). The horizontal mirroring, which transformed the left and right sides of the image centering on the vertical line of the image, was achieved by setting the flip parameter 'dim' to '1'. The vertical mirroring, which transformed the upper and lower sides of the image centering on the horizontal centerline of the image, was achieved by setting the flip parameter 'dim' to '2'. For image rotation, the Matlab function "imrotate" was used to rotate the raw image, and 90°, 180°, and 270° of rotation were achieved by changing the function parameter 'angle', respectively. The transformed images can improve the detection performance of the neural network by correctly identifying the apples of different orientations.

The image enhancement methods adopted in this study were brightness variation, histogram equalization and image blurry. The brightness variation was applied five times to enhance the illumination range of the raw training datasets. Multiplying a proportional coefficient near 1.0 by the original image RGB can adjust the value of each component to make the image brightness higher or lower (Tian et al.,

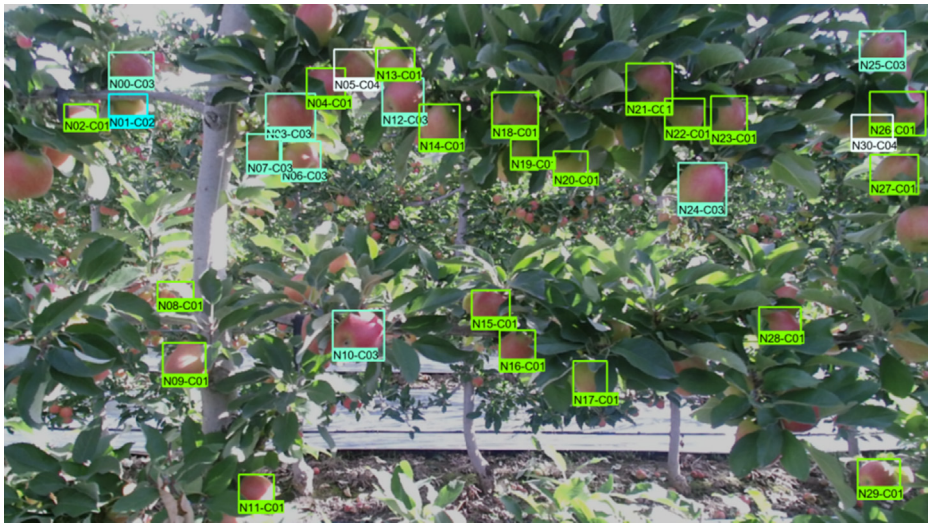


Fig. 6. Ground-truth apple targets were manually annotated using rectangular annotations. C01, C02, C03, and C04 refer to the leaf-occluded, branch/wire-occluded, non-occluded, and fruit-occluded fruit, individually. The number after N means the number of fruits being labeled in this image.

2019). 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. Five proportional coefficients of 0.7, 0.8, 0.9, 1.1, 1.2 were selected based on the target edge which can be accurately identified during manual annotation. If the multiplied value was higher than 255, it was automatically adjusted to 255.

The histogram equalization method was used to improve the quality of the training sample images and the variety of illuminations. The original RGB (Red, Green, Blue) color image was converted to HSV (Hue, Saturation, Value) color space using the Matlab function 'rgb2hsv' (Smith, 1978). Then the histogram equalization was performed on the V component of the HSV using the Matlab function 'histeq' with default parameters (Hou and Liu, 2017). After that, the new V component with original H and S were converted back to the RGB color image using the Matlab function 'hsv2rgb' (Smith, 1978), which was employed as the augmented image by the histogram equalization method.

The image blurry was employed four times to make the convolutional network model have strong adaptability to blurred images (caused by moving cameras). A predetermined two-dimensional filter was created using the Matlab function 'fspecial' (Tani et al., 2016). Since the telephoto distance of the camera, incorrect focusing, and camera movement will cause blurry images that are difficult to estimate, parameters LEN and THETA of the Motion filter are determined. LEN (*length*, represents pixels of linear motion of camera) and THETA (θ, represents the angular degree in a counter-clockwise direction) were set as (20, −15), (20, 15), (30, −20) and (30, 20), respectively. Then, the Matlab function 'imfilter' is used to blur the image with the generated filter.

The raw training datasets were augmented 15 times by the above methods, and the number of images in the dataset was increased to 12,800 (including the raw datasets) from 800, which was divided into training (7680 images, 60%), validation (2560 images, 20%), and testing (2560 images, 20%) datasets.

The 'xml' file of the original image needs to be modified after the image was augmented by the geometric transformation. The values of *xmin* and *ymin* in the 'xml' file represent the coordinates of the upper-left vertex of the rectangular annotation box in the image. The values of *xmax* and *ymax* in the 'xml' file represent the coordinates of the bottom-right vertex of the rectangular annotation box. The geometric transformation operation changed the position of the fruit on the image, while the image enhancement operation didn't change the position of the fruit in the image. When the position of fruits changed, the coordinate values (*xmin*, *ymin*) and (*xmax*, *ymax*) of the rectangular annotation boxes need to be recalculated based on the specific performed transformation. The related data augmentation processes are shown in Fig. 7. The dataset with corresponding annotations has been made publicly available on github.com (https://github.com/fu3lab/Scifresh-apple-RGB-images-with-multi-class-label).

2.4. Deep learning model

Although Faster R-CNN has been widely applied in the field of multi-target fruit detection and achieved promising results, different class of apples still need to be separately detected for improving the performance of the robot. Faster R-CNN model merges region proposals network (RPN), object classification and object localization into one unified deep object detection network, and share the same convolution features, as shown in Fig. 8. The RPN is used to generate the proposals, and the Fast R-CNN is used to accurately locate the object (Ren et al., 2017). The RPN, a fully convolutional neural network, uses a partial convolutional layer of VGG16 network to generate a feature map of an apple image and outputs a series of apple target candidate regions (Abdalla et al., 2019). In order to generate the apple target candidate regions, an $n \times n$ sliding window is used to scan the feature map of the apple image, and m target candidate regions are predicted for the position of each sliding window. The m proposals for the same localization are called anchors. An anchor point is located in the center of the sliding window and related to scale and aspect ratio. By default, 3 scales and 3 aspect ratios are used to generate $m = 9$ anchors. Two fully connected layers of the same level regression layer and classification layer are following 512-dimensional features (Zhang et al., 2018a,b). A regression layer was used to predict center coordinates and the aspect ratio of the anchor and the classification layer was used to judge whether the proposal is an object or the background.

VGG16 won the second place in 2014 ILSVRC (ImageNet Large Scale Visual Recognition Challenge) and performed well in multiple transfer learning tasks. There are 16 convolutional layers and fully connected layers in this network, with 13 shareable convolutional layers and three fully connected layers (Hossain et al., 2019; Li et al., 2020; Zhang et al., 2020a,b). The RGB image was used as the input of VGG16 and then processed by the network. The network performs a random gradient descent method on the blocks of the image to update the parameters. A filter of size 3×3 with stride 1 filter was used to construct a convolutional layer by VGG16, where the padding parameter in the same convolution is used as its parameter. Then a 2×2 with stride 2 filter was used to build the max-pooling layer. The feature map of image was extracted through the convolution, ReLu (rectified linear unit) and pooling operations, which was shared in the subsequent RPN layers and fully connected layers.

ZFNet is another commonly used network in sharing convolution between the RPN and the Faster R-CNN. ZFNet was the winner of the ILSVRC in 2013, which consists of five shareable convolutional layers, max-pooling layers, dropout layers, and three fully connected layers. ZFNet is also known for its high detection rate and fast speed, which has been verified previously (Yang et al., 2018; Bian et al., 2019). Therefore, ZFNet was selected for comparison with the VGG16 in this study. By comparing these two networks with different depths, the effect of the depth of the network on the fruit detection results is discussed in this study.

2.5. Network training

The training process for apple detection is based on Faster R-CNN with VGG16. The RPN was implemented as a full convolutional network that was optimized through an end-to-end using backpropagation and mini-batch gradient descent (Fuentes et al., 2017). In training, the loss function is comprised of the sum of a classification loss and a bounding box regression loss. The proposal was considered positive detections if Intersection over Union (IoU) is greater than 0.5 in this work.

The specific steps of training were as the following. A fixed value of 0.9 was set as momentum of the network, and 0.0005 was used as weight decay. Mini-batches of 128 images were used to train the model. The constant learning rate of 0.001 was used for all layers in the network. Iterations of 100,000 were selected in order to analyze the training process. The training platform included a computer with Intel Xeon E5-1650 (3.60 GHz) six-core CPU, and a GPU of NVidia TITAN XP 6 GB GPU (3840 CUDA cores) and 16 GB of memory, running on a Windows 7 64-bit system. The software tools included CUDA 8.1, CUDNN 7.5, Python 2.7, and Microsoft Visual Studio 12.0. The experiments were implemented in the TensorFlow framework. The same training parameters and platform were applied to ZFNet for comparison. And the detection speed was also measured in the same computer hardware.

2.6. Network evaluations

Evaluation indicators were used to assess the performance of the trained model on testing dataset. For this study, all samples were divided into four types: true positive (*TP*), false positive (*FP*), true negative (*TN*), and false negative (*FN*), according to the combinations of

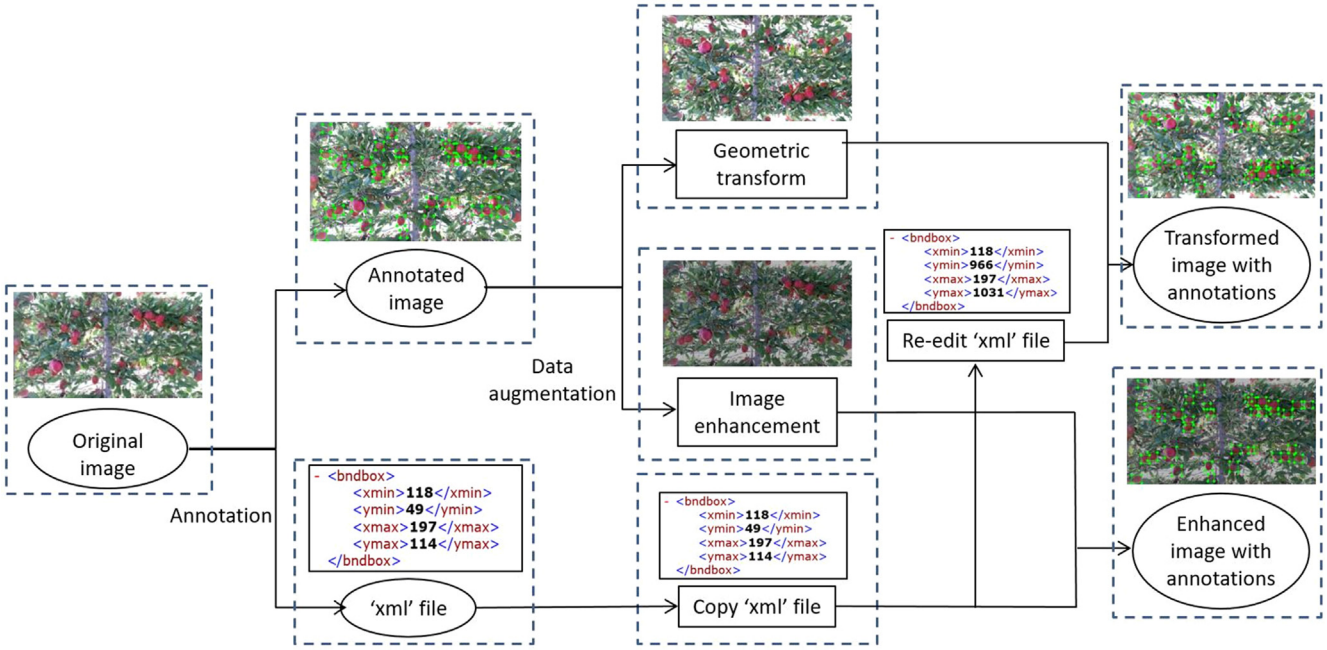


Fig. 7. The flow chart of data augmentation processes. The rectangular annotations of each image were saved in an 'xml' file. The data enhancement images only need to copy the 'xml' file of the original image for annotations. While the geometric transformed images need to re-edit the 'xml' file of the original image for annotations based on a specific transformed method, an example of the vertical mirror was shown in the Figure.

the true and predicted class. With these four types of samples, Precision (P) and recall (R) are defined in Eqs. (1) and (2), respectively.

$$P = \frac{TP}{TP + FP} \tag{1}$$

$$R = \frac{TP}{TP + FN} \tag{2}$$

The precision-recall curve (i.e., P-R curve) can be obtained by ploting the P as the vertical axis and the R as the horizontal axis. Since the evaluation index mainly focuses on the positive sample, thus to weigh the Precision index and the Recall index, AP_k was defined in Eq. (3) as the area under the P_k and R_k curve of the kth class. AP is a standard measure for measuring the sensitivity of the network to a target object, and is an indicator that reflects the global performance of the network. And mAP was defined in Eq. (4) as the average precision of the four classes of apples. The higher the AP and mAP, the better the detection results of the convolutional neural network for a given object (Zhang et al., 2020a,b). The value k represents the each class of fruits in the study: leaf-occluded fruit ($k = 1$), branch/wire-occluded fruit ($k = 2$), non-occluded fruit ($k = 3$), and fruit-occluded fruit ($k = 4$). The average detection time was also calculated to evaluate the performance of the model.

$$AP_k = \int_0^1 P(R_k)dR_k \tag{3}$$

$$mAP = \frac{1}{4}\sum_{k=1}^{4} AP_k \tag{4}$$

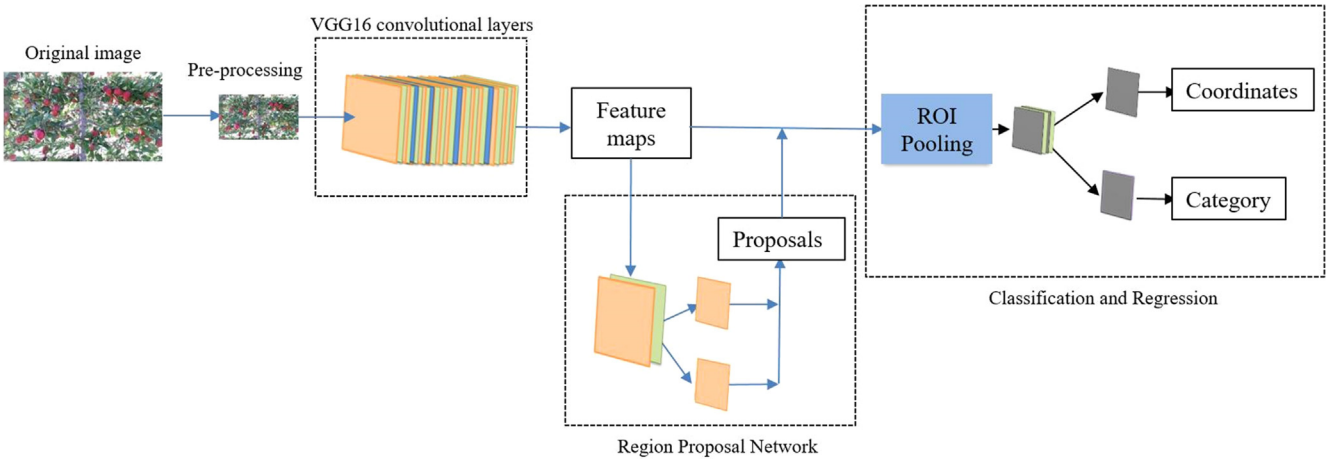


Fig. 8. Training architecture for apple detection based on Faster R-CNN with VGG16. The orange, blue, green and gray layers represent the convolutional layer, pooling layer, ReLU layer, and full connection layer, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

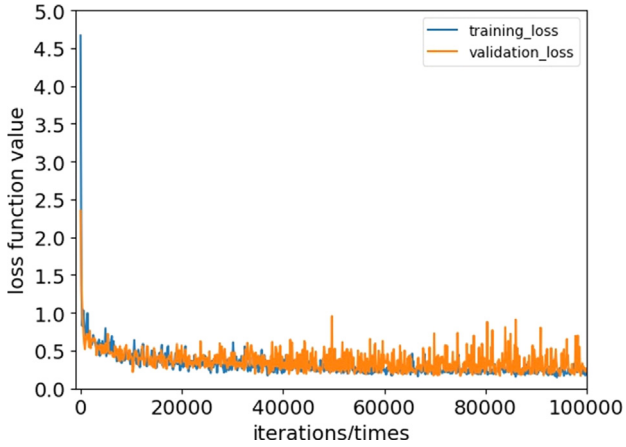


Fig. 9. Training and validation loss curve of VGG16 network.

3. Results and discussion

3.1. Training assessment and performance of the network

The training results of deep learning models are affected by the number of iterations. Fig. 9 illustrates the results of the loss curve of the training set and validation set for 100,000 iterations. The variation trend of training and validation datasets was almost overlapping, indicating that the model was not overfitted with the parameters selected in the validation process. The loss value decreases as the number of iterations increases, but is generally stable when the number of iterations reaches 70,000 iterations, and gradually approached the lowest value of 0.1631. The results demonstrate that the network adopted in this work efficiently learns the features with good convergence ability, which has the potential to achieve desired results.

The *P-R* curves achieved by Faster R-CNN based VGG16 and ZFNet on the testing dataset were shown in Fig. 10. As expected, *P* values of the non-occluded fruit was the highest of VGG16 and ZFNet with the same *R* values. Because the non-occluded fruit is fully visible and easier to be detected compared to those partially visible fruits (Nguyen et al., 2016). The leaf-occluded fruit achieved the second highest accuracy with both networks. The branch/wire-occluded fruit performed better than the fruit-occluded fruit in VGG16, while the result was reversed in ZFNet. On the other hand, the fruit-occluded fruit can become a non-occluded fruit in a subsequent detection once its outer fruit is picked (Koirala et al., 2019b). For the two networks, VGG16 obtained a higher *P* on all the four classes than that of ZFNet with the same *R* value.

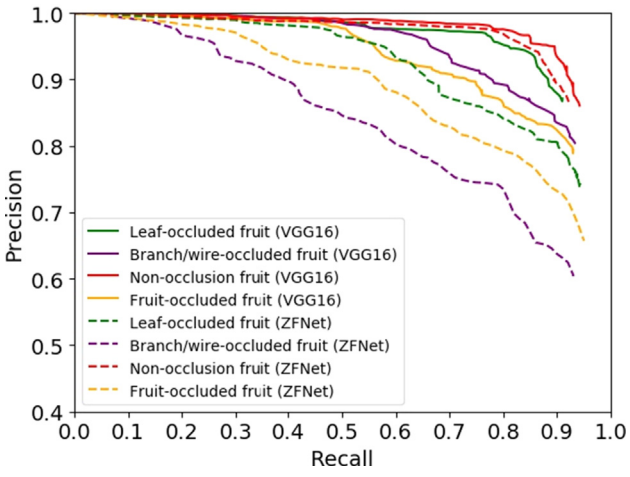


Fig. 10. Precision-Recall (P-R) curves of the models on the testing dataset with VGG16 (solid curves) comparing against ZFNet (dashed curves).

3.2. Multi-class apple detection with Faster R-CNN

Overall, VGG16 outperformed than ZFNet. The two architectures obtained the *mAP* values of 0.879 and 0.793, respectively, as shown in Table 1. The *mAP* of VGG16 is 8.6% higher than that of ZFNet using the same testing dataset. In terms of different classes, VGG16 and ZFNet both achieved the similar highest *AP* values of 0.909 and 0.902 on the non-occluded fruit, respectively. The second highest *AP* of VGG16 (0.899) and ZFNet (0.813) were observed in the class of leaf-occluded fruit. The branch/wire-occluded achieved a higher *AP* of 0.858 than the fruit-occluded fruit in VGG16, which was reversed in ZFNet. The *AP* of branch/wire-occluded fruit in the VGG16 is 14.5% higher than that of the same classes in ZFNet. The high-precision detection of apple fruits in this class using VGG16 provides promising results for the fruit robotic picking system. Although VGG16 achived the lowest *AP* of 0.848 on the fruit-occluded fruit, it can become a non-occluded fruit in a subsequent detection once its outer fruit is picked (Koirala et al., 2019b). Such results could also provide further information on picking strategy (e.g., picking order and path planning) for robotic system.

VGG16 took 0.241 s on average to detect apples in one image with the resolution of 1920 × 1080 pixels, which was about 1.5 times longer than ZFNet (0.167 s per image). The reason is that VGG16 has more convolutional layers and parameters than ZFNet that resulted in a large trained weights size of 512 MB, which is more than twice as of ZFNet (225 MB). The trained weights size in this study is similar to that on mango RGB images detection (Koirala et al., 2019b), which were 533 MB and 230 MB for VGG16 and ZFNet, respectively. For some of the current apple-picking robots that may need about 6 s on average to pick a fruit (Silwal et al., 2017), the detection speed achieved here deemed acceptable and could potentially be able to have a second round of detection after the target fruit is picked, particularly with fruit clusters.

The observation of bounding boxes on the resulting images presents more insightful results, as shown in Fig. 11. For the same image, more apple fruits were detected by VGG16 than ZFNet, for example, four apples were missed in Fig. 11a (VGG16), whereas six apples were missed in Fig. 11b (ZFNet), where the two more missed apples by ZFNet were both the leaf-occluded fruit. Although the same dataset was used to train the two different networks, the same fruits were detected as different classes, as shown by the black rectangle in Fig. 11. A fruit-occluded fruit was wrongly detected as non-occluded fruit in the black rectangle of Fig. 11a, while the same fruit was detected correctly in Fig. 11b. Furthermore, a leaf-occluded fruit was wrongly detected as non-occluded fruit in the black rectangle of Fig. 11b, while the same fruit was detected correctly in Fig. 11a.

For this work, the ultimate objective of the multi-class apple detection was to help the robot deciding picking strategy (e.g., picking order and path planning) as well as to avoid the potential damage by the branches and trellis wires. Most branch/wire-occluded fruit were correctly detected, as shown by the green rectangles in Fig. 11. Two branch/wire-occluded fruits were missed, as shown by the green circles in Fig. 11. However, these type of fruit will not be robotically picked using end-effector because none of them are wrongly detected as any other class. The same justification also can be used for the fruit behind the tube, as shown by the black circle in Fig. 11. It is missed in detection because it does not belong to any class and was not annotated. This type of fruit should also be avoided for robotic picking to protect the machine. The 'Scifresh' apple trees used in this work were trained in formal vertical architecture system, fruit detection and robotic harvesting could be conducted from both sides of the tree row. Therefore, the fruits that cannot be picked from one side are possible to be picked from the other side. By detecting the apple fruits in one tree as multi classes, it can avoid possible robot damage and economic losses caused by robotc picking on branch/wire-occluded fruits.

Table 1
Multi-class apple detection results with VGG16 and ZFNet.

	AP				*mAP*	Detection speed (s/image)
	Leaf-occluded fruit	Branch/wire-occluded fruit	Non-occluded fruit	Fruit-occluded fruit		
VGG16	0.899	0.858	0.909	0.848	0.879	0.241
ZFNet	0.813	0.713	0.902	0.743	0.793	0.167

3.3. Results from other studies on apple detection

Although there are no reported study on multi-class apple detection and it is difficult to compare methodologies tested with different datasets from different apple varieties, it still gives some insights when the results achieved by other studies using fruit images captured from different orchard environments are discussed along with the results achieved in this work. Especially the studies where the fruit occlusions were discussed (Lv et al., 2016; Nguyen et al., 2016; Feng et al., 2019; Liu et al., 2019; Wang and He, 2019). Results from these studies are summarized in Table 2. Wang and He (2019), Liu et al. (2019) and Lv et al. (2016) were working on apple orchards in China, which are traditional apple trees without training. Feng et al. (2019) and Nguyen et al. (2016) were working on apple trees that planted in tall spindle structure.

Most studies detected all the apple fruits as one class and divided them into different types to analyze the detection results for statistics. Lv et al. (2016) used random Hough transform (RHT) to detect 'Fuji' apples image with 1–5 fruits and reported 100% success rate on the non-occluded and fruit-occluded fruit, respectively. They reported detection rate of 86% on fruits occluded by branch and leaf together. Liu et al. (2019) also worked on 'Fuji' apples and analyzing the detected fruits in the same way. They employed linear iterative clustering (SLIC) and SVM to detect apples and obtained 94.8%, 85.8%, and 89.5% on the non-occluded, fruit-occluded, and branch and leaf occluded fruits, respectively. Lv et al. (2016) and Feng et al. (2019) presented the apple detection results by non-occluded and partially occluded (including branch-occluded, leaf-occluded, and fruit-occluded). Both achieved high success rates of 100% and 92% on the non-occluded fruit, respectively, while 82% and 72% were obtained on the partially occluded fruit. Wang and He (2019) labeled all the 'Fuji' apples as one class and applied region-based fully convolutional network (R-FCN) for detection, and then divided the detection results into four types for data analysis. They obtained detection rates of 91.6%, 75.1%, 78.8%, and 87.3% on the non-occluded, branch-occluded, leaf-occluded, and fruit-occluded fruit. Their detection results were ~2% higher on average than this research on the non-occluded and fruit-occluded fruit, but much lower (~10%) on the branch-occluded and leaf-occluded fruit. The reason might be that Wang and He (2019) trained all the apples as one class and learned the most features from the fruits other than the features of the branch and leaf that occluded the fruit, while this study trained the branch/wire-occluded and leaf-occluded fruits as separated classes. In this case, more features of the branch and leaf that occlude the fruit might be learned instead of only learning the features of the fruits. Lastly, none of those studies reported the wire-occluded fruit because that was neither included in their captured images nor considered by the researchers.

Based on Table 2, the state-of-art deep learning-based fruit detection methodology (Wang and He (2019) and our study) did not obtain better performances than the traditional image processing methods (Lv et al. (2016); Nguyen et al. (2016); Feng et al. (2019); Liu et al. (2019)) as expected. It was because those studies were working on simple apple images that does not have complex background. For those simple apple images, Kang and Chen (2020) and Tian et al. (2019) achieved nearly 100% detection rate using LedNet and YOLOv3, including fruits occluded by branches and leaves, as shown in Fig. 1. In the aspect of detection speed, the deep learning methods presented faster processing time than the traditional image processing methods, even they were working on higher resolution and more complicated images with more fruits.

4. Conclusions

The state-of-art deep learning technologies achieved fast speed and high success rate of fruit-on-plant detection in the orchard, but all the target fruits have been detected as one object class (apple), which includes not only fruit without the occasion or occluded by leaves but also fruit partly occluded by branches and trellis wires. These obstacles such as branches and wires might diminish the effectiveness of fruit picking and even damage the robotic end-effecter. In order to avoid the robotic end-effector from being damaged by these obstacles, a multi-class fruit-on-plant detection method for apples in modern orchards with fruiting-wall tree architecture was proposed based on Faster R-CNN. The major finding in this study are as followed.

Fig. 11. Examples of apple images detected by Faster R-CNN based VGG16 (a) and ZFNet (b). The rectangle of yellow, green, red and blue colors are referring to the detected leaf-occluded fruit, branch/wire-occluded fruit, non-occluded fruit, and fruit-occluded fruit, respectively. While circles in the corresponding color are manually added and indicating the missed fruit in each class. The black circles are manually added and indicating fruits that do not belong to any class, and the black rectangles are manually added and indicating the fruits that detected in the wrong class. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Results from previous studies on apple detection.

	Apple variety	Image size (pixels)	Fruits per image	Main method	Detection results of different classes				Detection speed (s/image)
					Non-occluded	Branch/wire-occluded	Leaf-occluded	Fruit-occluded	
Lv et al. (2016)	'Fuji'	320 × 240	1–5	RHT	100%	86%		100%	0.77
Nguyen et al. (2016)	'Fuji'	512 × 424	7–28	CHT, blob analysis	100%	82%			0.96
Liu et al. (2019)	'Fuji'	400 × 300	1–5	SLIC, SVM	94.8%	89.5%		85.8%	1.94
Feng et al. (2019)	'Crimson Crisp', 'Gold Rush'	400 × 300	2–10	GLCM, SVM	92%	72%			0.74
Wang and He (2019)	'Fuji'	500 × 500	10–15	R-FCN	91.6%	75.1%	78.8%	87.3%	0.187
Our method	'Scifresh'	1920 × 1080	35–50	VGG16	90.9%	85.8%	89.9%	84.8%	0.241

Note: RHT, random Hough transform; CHT, circular Hough transform; SLIC, linear iterative clustering; SVM, support vector machine; R-FCN, region-based fully convolutional network.

The fruits were labeled as four classes, i.e., non-occluded fruit, leaf-occluded fruit, branch/wire-occluded fruit, and fruit-occluded fruit, based on occlusion condtions of fruit robotic harvesing.

VGG16 network was used to implement the Faster R-CNN and achieved the *mAP* of 0.879 for the four classes.

The *AP* for the non-occluded fruit, leaf-occluded fruit, branch/wire-occluded fruit, and fruit-occluded fruit were 0.909, 0.899, 0.858, and 0.848, respectively.

Most branch/wire-occluded fruits were correctly detected and few of them were missed instead of wrongly detected as other class.

VGG16 took 0.241 s on average to detect apples on one image with the resolution of 1920 × 1080 pixels. This processing speed enables a fruit detection system to run at near real-time and may able to have a second round of detection after the fruit at the canopy surface are picked, since some of the current apple-picking robot need 6 s on average to pick one fruit. The results indicated that the branch/wire-occluded fruit that are unsuitable for robotic picking can be effectively detected from other apple classes, which can help the robot to decide the picking strategy (e.g., picking order and path planning) as well as to avoid the potential damage by the branches and trellis wires.

CRediT authorship contribution statement

Fangfang Gao: Data curation, Investigation, Writing - original draft. **Longsheng Fu:** Conceptualization, Data curation, Methodology, Supervision, Writing - review & editing. **Xin Zhang:** Conceptualization, Methodology, Writing - review & editing. **Yaqoob Majeed:** Conceptualization, Methodology, Writing - review & editing. **Rui Li:** Investigation, Supervision, Writing - review & editing. **Manoj Karkee:** Methodology, Writing - review & editing. **Qin Zhang:** Methodology, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors express their deep gratitude to the Young Faculty Study Abroad Program of the Northwest A&F University Scholarship who sponsored Dr Longsheng Fu in conducting post-doctoral research at the Centre for Precision and Automated Agricultural Systems, Washington State University, and to the Allan Brothers Fruit Company who provided the experimental orchard.

Funding

This work was supported by the China Postdoctoral Science Foundation funded project (2019M663832); Fundamental Research Funds for the Central Universities of China (2452020170); Key Research and Development Program in Shaanxi Province of China (grant number 2018TSCXL-NY-05-04, 2019ZDLNY02-04); National Natural Science Foundation of China (grant number 31971805); International Scientific and Technological Cooperation Foundation of Northwest A&F University (grant number A213021803).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compag.2020.105634.

References

Abdalla, A., Cen, H., Wan, L., Rashid, R., Weng, H., Zhou, W., He, Y., 2019. Fine-tuning convolutional neural network with transfer learning for semantic segmentation of ground-level oilseed rape images in a field with high weed pressure. Comput. Electron. Agric. 167, 105091. https://doi.org/10.1016/j.compag.2019.105091.

Bian, Y., Wang, J., Jun, J.J., Xie, X., 2019. Deep convolutional generative adversarial network (dcGAN) models for screening and design of small molecules targeting cannabinoid receptors. Mol. Pharm. 16, 4451–4460. https://doi.org/10.1021/acs.molpharmaceut.9b00500.

Feng, J., Zeng, L., He, L., 2019. Apple fruit recognition algorithm based on multi-spectral dynamic image analysis. Sensors 19, 0949. https://doi.org/10.3390/s19040949.

Fu, L., Tola, E., Al-Mallahi, A., Li, R., Cui, Y., 2019. A novel image processing algorithm to separate linearly clustered kiwifruits. Biosyst. Eng. 183, 184–195. https://doi.org/10.1016/j.biosystemseng.2019.04.024.

Fuentes, A., Yoon, S., Kim, S.C., Park, D.S., 2017. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors 17, 2022. https://doi.org/10.3390/s17092022.

Gené-Mola, J., Vilaplana, V., Rosell-Polo, J.R., Morros, J.R., Ruiz-Hidalgo, J., Gregorio, E., 2019. Multi-modal deep learning for Fuji apple detection using RGB-D cameras and their radiometric capabilities. Comput. Electron. Agric. 162, 689–698. https://doi.org/10.1016/j.compag.2019.05.016.

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., Lewis, K., 2015. Sensors and systems for fruit detection and localization: a review. Comput. Electron. Agric. 116, 8–19. https://doi.org/10.1016/j.compag.2015.05.021.

Häni, N., Roy, P., Isler, V., 2020. A comparative study of fruit detection and counting methods for yield mapping in apple orchards. J. F. Robot. 37, 263–282. https://doi.org/10.1002/rob.21902.

Hossain, M.S., Al-hammadi, M., Muhammad, G., 2019. Automatic fruit classification using deep learning for industrial applications. IEEE Trans. Ind. Inf. 15, 1027–1034. https://doi.org/10.1109/TII.2018.2875149.

Hou, J., Liu, W., 2017. A parameter-independent clustering framework. IEEE Trans. Ind. Inf. 13, 1825–1832. https://doi.org/10.1109/TII.2017.2656909.

Huang, L., Pan, W., Zhang, Y., Qian, L., Gao, N., Wu, Y., 2020. Data augmentation for deep learning-based radio modulation classification. IEEE Access 8, 1498–1506. https://doi.org/10.1109/ACCESS.2019.2960775.

Kang, H., Chen, C., 2020. Fast implementation of real-time fruit detection in apple orchards using deep learning. Comput. Electron. Agric. 168, 105108. https://doi.org/10.1016/j.compag.2019.105108.

Koirala, A., Walsh, K.B., Wang, Z., McCarthy, C.L., 2019a. Deep learning – Method overview and review of use for fruit detection and yield estimation. Comput. Electron. Agric. 162, 219–234. https://doi.org/10.1016/j.compag.2019.04.017.

Koirala, A., Walsh, K.B., Wang, Z., McCarthy, C.L., 2019b. Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'MangoYOLO'. Precis. Agric. 20, 1107–1135. https://doi.org/10.1007/s11119-019-09642-0.

Li, J., Xue, Y., Wang, W., Ouyang, G., 2020. Cross-level parallel network for crowd counting. IEEE Trans. Ind. Inf. 16, 566–576. https://doi.org/10.1109/TII.2019.2935244.

Lin, G., Tang, Y., Zou, X., Xiong, J., Fang, Y., 2020. Color-, depth-, and shape-based 3D fruit detection. Precis. Agric. 21, 1–17. https://doi.org/10.1007/s11119-019-09654-w.

Liu, Z., Wu, J., Fu, L., Majeed, Y., Feng, Y., Li, R., Cui, Y., 2020. Improved kiwifruit detection using pre-trained VGG16 with RGB and NIR information fusion. IEEE Access 8, 2327–2336. https://doi.org/10.1109/ACCESS.2019.2962513.

Liu, X., Zhao, D., Jia, W., Ji, W., Sun, Y., 2019. A detection method for apple fruits based on color and shape features. IEEE Access 7, 67923–67933. https://doi.org/10.1109/ACCESS.2019.2918313.

Luus, F.P.S., Salmon, B.P., Van den Bergh, F., Maharaj, B.T.J., 2015. Multiview deep learning for land-use classification. IEEE Geosci. Remote Sens. Lett. 12, 2448–2452. https://doi.org/10.1109/LGRS.2015.2483680.

Lv, J., Zhao, D., Ji, W., Ding, S., 2016. Recognition of apple fruit in natural environment. Optik (Stuttg). 127, 1354–1362. https://doi.org/10.1016/j.ijleo.2015.10.177.

Majeed, Y., Zhang, J., Zhang, X., Fu, L., Karkee, M., Zhang, Q., 2020. Deep learning based segmentation for automated training of apple trees on trellis wires. Comput. Electron. Agric. 170, 105277. https://doi.org/10.1016/j.compag.2020.105277.

Nguyen, T.T., Vandevoorde, K., Wouters, N., Kayacan, E., De Baerdemaeker, J.G., Saeys, W., 2016. Detection of red and bicoloured apples on tree with an RGB-D camera. Biosyst. Eng. 146, 33–44. https://doi.org/10.1016/j.biosystemseng.2016.01.007.

Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39, 1137–1149. https://doi.org/10.2307/j.ctt1d98bxx.10.

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., McCool, C., 2016. Deepfruits: a fruit detection system using deep neural networks. Sensors 16, 2019. https://doi.org/10.3390/s16081222.

Silwal, A., Davidson, J.R., Karkee, M., Mo, C., Zhang, Q., Lewis, K., 2017. Design, integration, and field evaluation of a robotic apple harvester. J. F. Robot. 34, 1140–1159. https://doi.org/10.1002/rob.21715.

Smith, A.R., 1978. Color gamut transform pairs. Comput. Graph. 12, 12–19. https://doi.org/10.1145/965139.807361.

Tang, Y., Chen, M., Wang, C., Luo, L., Li, J., Lian, G., Zou, X., 2020. Recognition and localization methods for vision-based fruit picking robots: a review. Front. Plant Sci. 11, 510. https://doi.org/10.3389/fpls.2020.00510.

Tani, J., Mishra, S., Wen, J.T., 2016. Motion blur-based state estimation. IEEE Trans. Control Syst. Technol. 24, 1012–1019. https://doi.org/10.1109/TCST.2015.2473004.

Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., Liang, Z., 2019. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Comput. Electron. Agric. 157, 417–426. https://doi.org/10.1016/j.compag.2019.01.012.

Wan, S., Goudos, S., 2020. Faster R-CNN for multi-class fruit detection using a robotic vision system. Comput. Netw. 168, 107036. https://doi.org/10.1016/j.comnet.2019.107036.

Wang, D., He, D., 2019. Recognition of apple targets before fruits thinning by robot based on R-FCN deep convolution neural network. Trans. Chinese Soc. Agric. Eng. 35, 156–163. https://doi.org/10.11975/j.issn.1002-6819.2019.03.020.

Wang, Y., Li, W., Xu, X., Qiu, C., Wu, T., Wei, Q., Ma, F., Han, Z., 2019. Progress of apple rootstock breeding and its use. Hortic. Plant J. 5, 183–191. https://doi.org/10.1016/j.hpj.2019.06.001.

Yang, T., Long, X., Sangaiah, A.K., Zheng, Z., Tong, C., 2018. Deep detection network for real-life traffic sign in vehicular networks. Comput. Networks 136, 95–104. https://doi.org/10.1016/j.comnet.2018.02.026.

Zhang, Z., Flores, P., Igathinathane, C., Naik, D.L., Kiran, R., Ransom, J.K., 2020b. Wheat lodging detection from UAS imagery using machine learning algorithms. Remote Sens. 12, 1838. https://doi.org/10.3390/rs12111838.

Zhang, J., He, L., Karkee, M., Zhang, Q., Zhang, X., Gao, Z., 2018a. Branch detection for apple trees trained in fruiting wall architecture using depth features and Regions-Convolutional Neural Network (R-CNN). Comput. Electron. Agric. 155, 386–393. https://doi.org/10.1016/j.compag.2018.10.029.

Zhang, Z., Heinemann, P.H., Liu, J., Baugher, T.A., Schupp, J.R., 2016. The development of mechanical apple harvesting technology: a review. Trans. ASABE 59, 1165–1180. https://doi.org/10.13031/trans.59.11737.

Zhang, J., Karkee, M., Zhang, Q., Zhang, X., Yaqoob, M., Fu, L., Wang, S., 2020a. Multi-class object detection using faster R-CNN and estimation of shaking locations for automated shake-and-catch apple harvesting. Comput. Electron. Agric. 173, 105384. https://doi.org/10.1016/j.compag.2020.105384.

Zhang, Z., Pothula, A.K., Lu, R., 2018b. A review of bin filling technologies for apple harvest and postharvest handling. Appl. Eng. Agric. 34, 687–703. https://doi.org/10.13031/aea.12827.

Zhao, D., Lv, J., Ji, W., Zhang, Y., Chen, Y., 2011. Design and control of an apple harvesting robot. Biosyst. Eng. 110, 112–122. https://doi.org/10.1016/j.biosystemseng.2011.07.005.