Received: 9 May 2020 | Revised: 28 October 2020 | Accepted: 29 October 2020

DOI: 10.1002/rob.21998

REGULAR ARTICLE

WILEY

Computer vision-based tree trunk and branch identification and shaking points detection in Dense-Foliage canopy for automated harvesting of apples

Xin Zhang[1,2] | Manoj Karkee[1,2] | Qin Zhang[1,2] | Matthew D. Whiting[2,3]

[1]Department of Biological Systems Engineering, Washington State University, Pullman, WA, USA

[2]Center for Precision and Automated Agricultural Systems, Washington State University, Prosser, WA, USA

[3]Department of Horticulture, Washington State University, Prosser, WA, USA

Correspondence
Manoj Karkee, Department of Biological Systems Engineering, Washington State University, Pullman, WA 99164, USA.
Email: manoj.karkee@wsu.edu

Funding information
National Institute of Food and Agriculture, Grant/Award Numbers: 1001246, 1005200, 1005756; USDA Hatch and Multistate Project Funds, Grant/Award Numbers: 1005756, 1001246; USDA National Institute for Food and Agriculture (NIFA), Grant/Award Number: 1005200; WSU Agricultural Research Center (ARC)

Abstract

Fresh market apples are one of the high-value crops in the United States. Washington alone has produced two-thirds of the annual national production in the past 10 years. However, the availability of seasonal labor is increasingly uncertain. Shake-and-catch automated harvesting solutions have, therefore, become attractive for addressing this challenge. One of the significant challenges in applying this harvesting system is effectively positioning the end-effector at appropriate excitation locations. A computer vision system was used for automatically identifying appropriate locations. Convolutional neural networks (CNNs) were utilized to identify the tree trunks and branches for supporting the automated excitation locations determination. Three CNN architectures were employed: Deeplab v3+ ResNet-18, VGG-16, and VGG-19. Four pixel classes were predefined as branches, trunks, apples, and leaves to segment the canopies trained to simple, narrow, accessible, and productive tree architectures with varying foliage density. Results on Fuji cultivar showed that ResNet-18 outperformed the VGGs in identifying branches and trunks based on all three evaluation measures: per-class accuracy (PcA), intersection over union (IoU), and boundary-F1 score (BFScore). ResNet-18 achieved a PcA of 97%, IoU of 0.69, and BFScore of 0.89. The ResNet-18 was further evaluated for its robustness with other test canopy images. When applied this method to one of the highest density cultivars of Scifresh, results showed it can achieve IoUs of 0.41 and 0.62 and BFScores of 0.71 and 0.86 for branches and trunks. Such identification result helped to get a 72% of auto-determined shaking points being the "good" category identified by human experts.

KEYWORDS

automated detection, convolutional neural networks, deep learning, network generalization, semantic segmentation, shake-and-catch harvesting, trajectory estimation

1 | INTRODUCTION

Fresh market apples are one of the high-value agricultural commodities in the United States and Washington State. About 300 thousand acres of apple, ~5.2 billion kilograms, are manually harvested each year nationally (USDA, 2020). However, the agricultural labor availability in the entire Pacific Northwest region and around the world has been increasingly uncertain, thus posing a considerable risk for the sustainable apple industry. For example, up to 100 million kilograms of apples were unharvested because of labor shortage during the 2007 and 2014 harvest seasons in Washington State (USDA, 2020). Besides, about 21% of Washington farms lost up to

$250,000 because of the same reason in 2016 (Clark, 2017). Therefore, apple growers in Washington State have a growing desire to consider adopting labor-saving harvest technologies including selective robotic pickers and vibratory shake-and-catch harvesters. A robotic apple picker requires the integration of many different components into a complex machine to selectively pick apples. Also, most of the currently developed robotic apple harvesting systems (e.g., Hohimer et al., 2019; Silwal et al., 2017) are very expensive, thus raising affordability concerns for commercial adoption. In addition, robotic picking systems are limited because of the generally low-harvesting efficiency they achieve (Bac et al., 2014). In comparison, shake-and-catch apple harvesting systems show promise in addressing these challenges, and it is expected that shake-and-catch harvesting could be more economically affordable and technically feasible for in-field applications.

Numerous studies on apple shake-and-catch harvesting have been previously conducted. First, tree trunk shaking with impacting actuators were investigated because the energy could be easily transferred to detach fruits (Peterson & Wolford, 2003). Later, studies were conducted to target tree branches for controlled shaking facilitated by simple, narrow, accessible, and productive (SNAP) fruiting-wall canopy architectures, where branches are often trained firmly onto horizontal trellis wires. With this canopy architecture, fruit catching systems can be inserted into the canopies to minimize the fruit dropping distance (Figure 1) and potential fruit damage. For instance, He et al. (2019) investigated a multilayer vibratory shake-and-catch harvester on Scifresh apples with optimal shaking location and duration. Overall, 85% of fruit removal efficiency was achieved using this technique. This study also verified that locally shaking at the base of the branches (close to the trunk) was approximately two times more efficient than shaking at the middle of the branches. Among the harvested apples, about 88% were reported to be marketable according to the United States Department of Agriculture (USDA) fruit quality standards (USDA, 2002). Although some of the latest studies on shake-and-catch harvesting show promising results in fruit detachment efficiency and fruit quality, these machines rely on manual operation, leading to inefficient and laborious maneuvering in the field. For example, the results show that the time spent positioning the actuator and shaking head into the canopy is almost eight times more than the time spent actuating the shaker. Especially with medium/high-vigor apple rootstocks in high-density SNAP tree architectures, high-density foliage might develop in canopies, making the localizing and accessing the targeted branch locations challenging because of occlusions caused by foliage and/or fruit and the fixed branch and tree spacing.

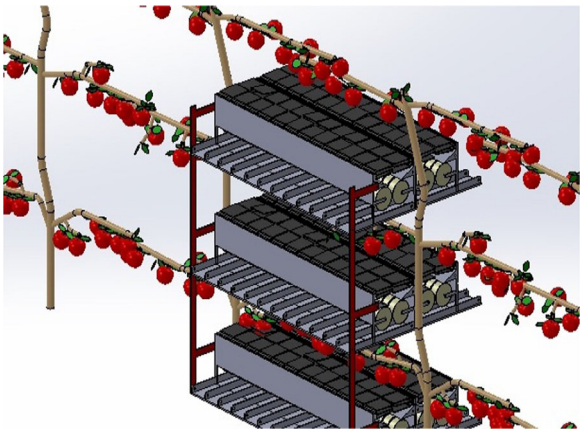

FIGURE 1 Conceptual illustration of a multilayer shake-and-catch harvesting system in a simple, narrow, accessible, and productive trellis-trained apple tree architecture [Color figure can be viewed at wileyonlinelibrary.com]

Because of such canopy conditions, most current shake-and-catch vibratory harvesting prototypes require field workers manually determining and engaging the machine to appropriate shaking positions for completing the harvesting task. It is laborious and could also induce some health risks for workers, such as inhaling dust when the machine was actuating vibration. To address this challenge, Peterson et al. (1999) investigated a robotic mass harvesting system for fresh-market apples and reported a fruit removal efficiency of 95%. However, a human operator was heavily involved in manually selecting the target locations on every captured image. Additional studies to further advance such harvesting systems using automated shaking location detection and engagement are therefore essential.

The first step in automating shake-and-catch harvesting systems is providing the capability for a harvester to automatically detect optimal shaking point(s) on target branches using computer vision techniques. A few studies have examined the process of locating the branches of trellis-trained cherry trees (Amatya et al., 2016, 2017), and their results showed an accuracy of 89% in detecting and locating branches in cherry trees. Briefly, their studies used conventional methods of machine vision systems for segmenting the cherry tree branches out, but these methods are generally less robust than deep learning techniques. Also, the tree architecture and foliage type they used was different; therefore, the findings would not be directly applicable to identifying target branches of apple trees. Moreover, some works were conducted to reconstruct three-dimensional (3D) tree trunks/branches for automated pruning operations (Karkee & Adhikari, 2015; Karkee et al., 2014). We do not further discuss the results because the application environment is entirely different. Specifically, pruning is normally conducted in the dormant season when no leaves are in the tree canopies. However, we need to identify the individual branches during the foliage season for harvesting purposes. In addition, the goal of developing such a machine vision system in this study is for real-time applications in orchard environments. Therefore, our study proposes a machine vision system that includes a convolutional neural network (CNN)-based image processing technique for the near-real-time automated detection of shaking locations during the foliage season in apple canopies.

Recently, deep learning-based technologies have been widely used to process images for agricultural applications because of their higher accuracy and robustness than most conventional algorithms (Kamilaris & Prenafeta-Boldú, 2018). CNNs are some of the most applied deep learning techniques because of their capabilities of

processing high-resolution image data and reasonable computational time made possible by network weight sharing among numerous convolutional layers.

There are two main types of CNN-based learning applications in agriculture, image semantic/instance segmentation and object detection (Chen et al., 2018; He et al., 2017; Ren et al., 2015). Many studies in the agricultural field have been conducted using object detection techniques. For example, Bargoti and Underwood (2017) used a Faster R-CNN-based object detection framework to detect various fruit types using color imaging techniques. The study showed promising results with a boundary-F1 score (BFScore) greater than 0.9 for apples and mangoes. Such detection systems are comparatively more robust and can potentially detect other similar objects under different cropping and environmental conditions. In contrast, semantic/instance segmentation methods have been more frequently used in analyzing remote sensing images such as satellite and unmanned aerial vehicle (UAV)-based images (Kemker et al., 2017; Sa et al., 2017). For ground vehicle use, Dias et al. (2018) presented a study on identifying multiple species of fruit flowers using a fully convolutional network (FCN).

Only limited studies have been previously reported on identifying tree trunks and branches for automated harvesting with shake-and-catch systems. Zhang et al. (2018) adopted an R-CNN-based object detection technique to detect the visible parts of apple tree branches in tree canopies trained to SNAP tree architecture. With the modification of a pretrained AlexNet (Krizhevsky et al., 2012), a deep learning architecture where the network has already been trained with informative features from an image data set such as the ImageNet (Deng et al., 2009), branch skeletons or trajectories were generated for an automated localization with the average recall of 92% and accuracy of 86%. However, this study was conducted in the dormant season and needed to be further improved for practical use during the harvesting season. Also, Majeed et al. (2020) employed a pretrained SegNet architecture to segment tree trunks and branches from the background with mean BFScores of 0.93 and 0.88 for trunks and branches, respectively. The study was also conducted in a dormant season with young 1-year-old apple trees. Gao et al. (2020) reported multiclass object detection on apples, branches, and trunks under full foliage conditions using Faster R-CNN. However, their work was not optimized for detecting branches with different cultivars for estimating shaking locations.

The primary goal of this study is to identify and locate the tree branches/trunks precisely and estimate suitable shaking locations in dense-foliage canopies for automating shake-and-catch harvesting systems for apples. The following are the specific objectives pursued:

(i) To automatically segment the tree trunks and branches using three different pretrained CNNs: Deeplab v3+ ResNet-18, and two SegNets: Visual Geometry Group (VGG)-16 and VGG-19;
(ii) To develop and implement a strategy for detecting shaking points on individual branches for automated mass harvesting.

2 | MATERIALS AND METHODS

2.1 | Experimental orchards

This study was conducted using SNAP trellis-trained apple trees in V-axis (Figure 2a) and vertical-axis (Figure 2b) architectures. The experiments were conducted in a commercial apple orchard near Prosser, WA, during the 2017–2018 harvesting seasons. Both architectures are currently widely used by growers in the Pacific Northwest region of the United States because of their uniform canopy light distribution, high fruit load, and good accessibility for humans and machines (Whiting, 2018). Tree trunks were trained to trellis wires with the elevation angle of 70° and 90° to the ground for V-axis or vertical-axis systems, respectively. Tree branches were horizontally trained along 7 to 8 trellis wires spaced about 0.5 m apart. Three different levels of foliage density because of the different vigor of rootstock were involved: light-density foliage canopy of Pink Lady, medium-density foliage canopy of Fuji, and high-density foliage canopy of Envy and Scifresh. The primary data collection was with Fuji; three other cultivars were involved in testing the performance in the situation outside of the training process. In these

FIGURE 2 Example of simple, narrow, accessible, and productive trellis-trained apple orchards in V-axis (a) and vertical-axis (b) architectures (Prosser, WA) [Color figure can be viewed at wileyonlinelibrary.com]

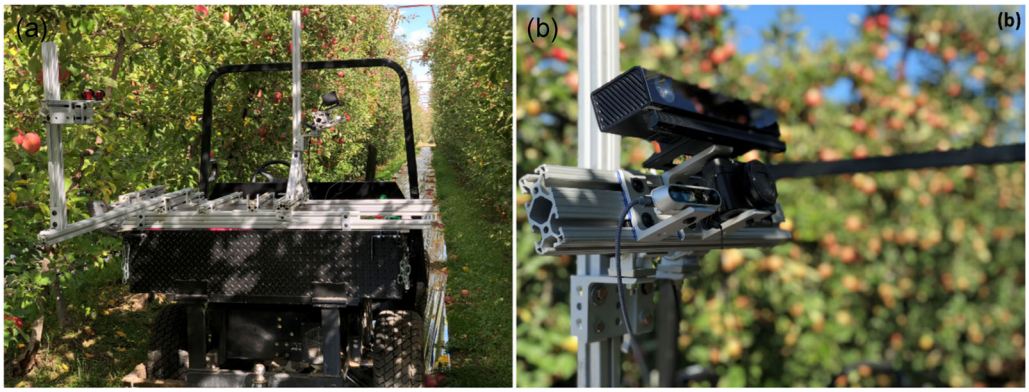

FIGURE 3 A customized image acquisition platform mounted on a Toro utility vehicle in a field environment (a); and closeup of the imaging system set up in an inclination such that it faces the V-axis canopies orthogonally (b) [Color figure can be viewed at wileyonlinelibrary.com]

orchards, crop and canopy structures are regularly maintained by laborers through training, pruning, and thinning. The description of the harvesting system (Figure 1) tests with those four commercial orchards can be found in Zhang et al. (2020).

2.2 | Image acquisition

A Kinect imaging sensor (Kinect V2, Microsoft Inc.) that consisted of red-green-blue (RGB), depth, and infrared channels was used (Figure 3). This sensor is relatively stable in the outdoor environment and is economically affordable. The RGB images are helpful in semantic segmentation with color and other associated features. The depth camera uses the time-of-flight principle with an infrared laser. It records 2.5-D information (i.e., point cloud data) that can be used to extract the locations of desired objects. The maximum effective pixel resolution for the RGB sensor is 1920 × 1080 and for the depth sensor is 512 × 424. The RGB and depth channels were coregistered together as the RGB-depth (RGB-D) image using a Kinect Software Development Kit (SDK) built-in function during a data acquisition operation. During the coregistration process, the depth channel (lower resolution) was mapped to the RGB channel (higher resolution) to ensure better pixel preservation. A customized platform mounted on an electric field vehicle was used for image acquisition (Figure 3a). The camera was orthogonally positioned to the target canopies (Figure 3b). The distance from the camera to the center of the target canopies was maintained between 1.1 and 1.2 m. The mobile platform was stationary when the images were acquired. A total of 785 canopy images were acquired under natural illumination conditions.

2.3 | Image pre-processing

Once the point cloud data (Figure 4a) were acquired, a few preprocessing techniques were applied before using them as inputs to CNNs. Figure 4b illustrates an example of an RGB image of the apple canopies used in this study. Because interrow spacing is 2.7–3.8 m, a depth threshold of 1.4–1.9 m (half of the row spacing) was used to remove objects from the adjacent rows and create a foreground image (Figure 4c). Our preliminary study showed that by filtering out the image background, the accuracy in object detection could be improved by ~2.5%, in general, compared to the images with background using deep learning techniques (Fu et al., 2020). The images were processed using a MATLAB (R2018b) software package on a Windows 10 (64-bit) platform with Intel Core i7-8750H CPU (2.20 GHz, 32.0 GB RAM, NVIDIA GeForce GTX 1080 GPU with Max-Q design). Both the original (1920 × 1080) and the resized (960 × 540) images were used. The contrasts of the foreground images were slightly enhanced using histogram equalization (Figure 4d). Images were then manually annotated using a MATLAB built-in tool (Image Labeler) to group pixels into one of four ground-truth image classes; (i) tree branches (in yellow), (ii) apples (in red), (iii) tree trunks (in white), and (iv) background (mostly leaves in purple; Figure 4e); and pixel-labeled images, where every pixel value represents a categorical label of that pixel. The varying level of exposure to sunlight in different parts of the canopy resulted in a variation in the brightness of leaves (lighter in the upper part and darker in the lower part of the canopy) during image data collection (Figure 4b). This variation was visually enhanced by the contrast-enhancement process using histogram equalization (Figure 4d). The final result of this study was not affected by the brightness variation. Data distribution of class labels in the full data set shows that leaves covered 91.41% of the area/pixels, and this was much more than the other three classes including 1.15% for branches, 6.20% for apples, and 1.25% for trunks. Therefore, median frequency class weights were calculated with 3.24 for branches, 0.60 for apples, 0.04 for leaves, and 2.99 for trunks and then reassigned to each class to balance the pixels difference.

2.4 | Semantic segmentation using deep learning

2.4.1 | CNNs architecture and activation channels

In this study, deep learning networks of semantic segmentation were employed. Three pretrained encoder-decoder architectures of CNNs

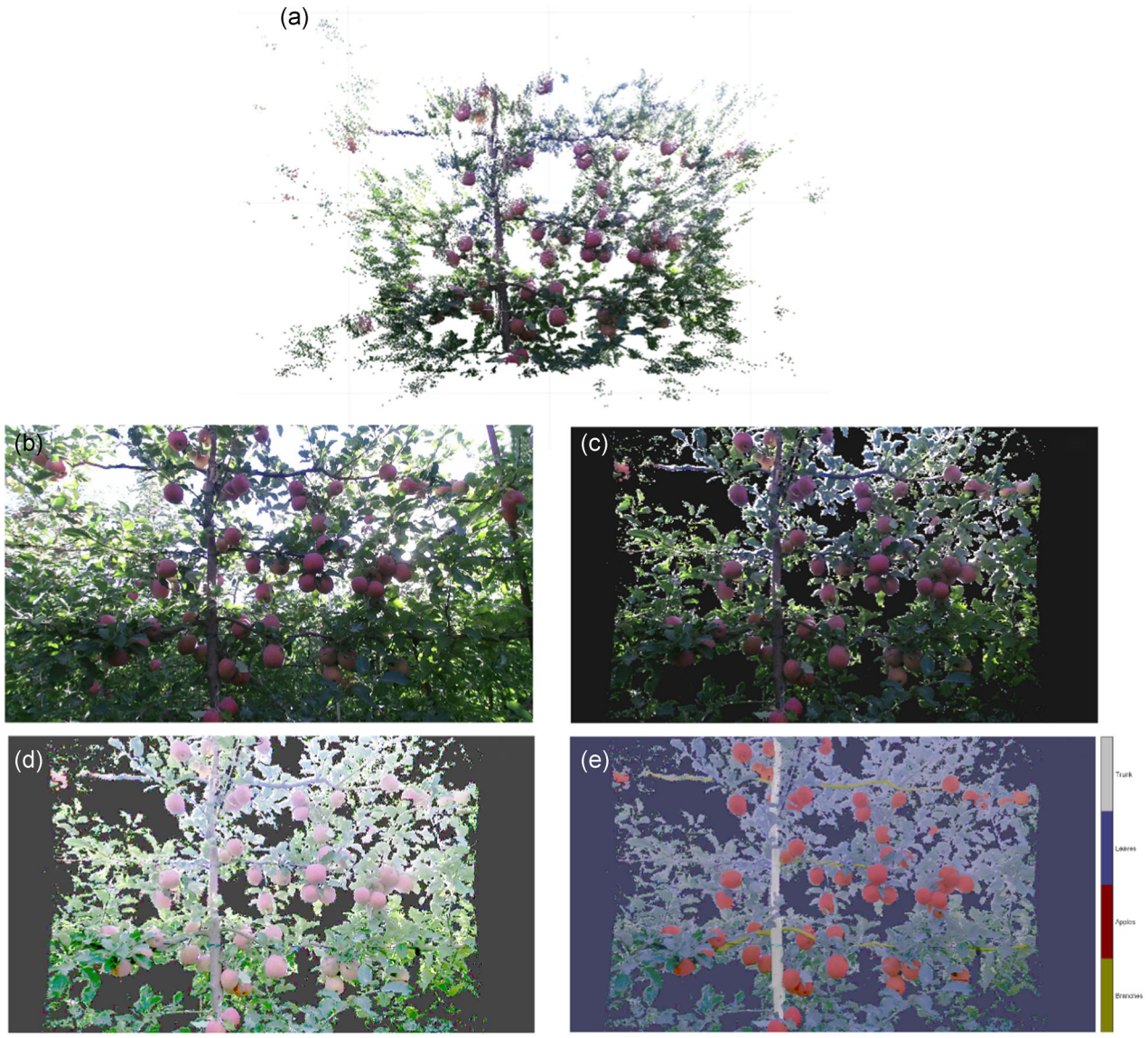

FIGURE 4 The illustration with medium-density foliage canopy of Fuji of canopy point cloud data (a); RGB image (b); foreground image after a depth threshold of 1.9 m was applied (c); contrast-enhanced image using histogram equalization (d); and corresponding pixel-wise segmented ground-truth image (e), where trunks are presented in white, leaves (background) are presented in purple, apples are presented in red, and branches are presented in yellow [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 1 Comparisons of the pretrained original and modified convolutional neural networks

Networks	Parameters	Original	Modified
Deeplab v3+ ResNet-18	Type	DAG network	DAG network
	Layer number	72	101
	Node connections	79	114
VGG-16	Type	Series network	DAG network
	Layer number	41	91
	Node connections	40	100
VGG-19	Type	Series network	DAG network
	Layer number	47	109
	Node connections	46	118

Abbreviation: DAG, directed acyclic graph.

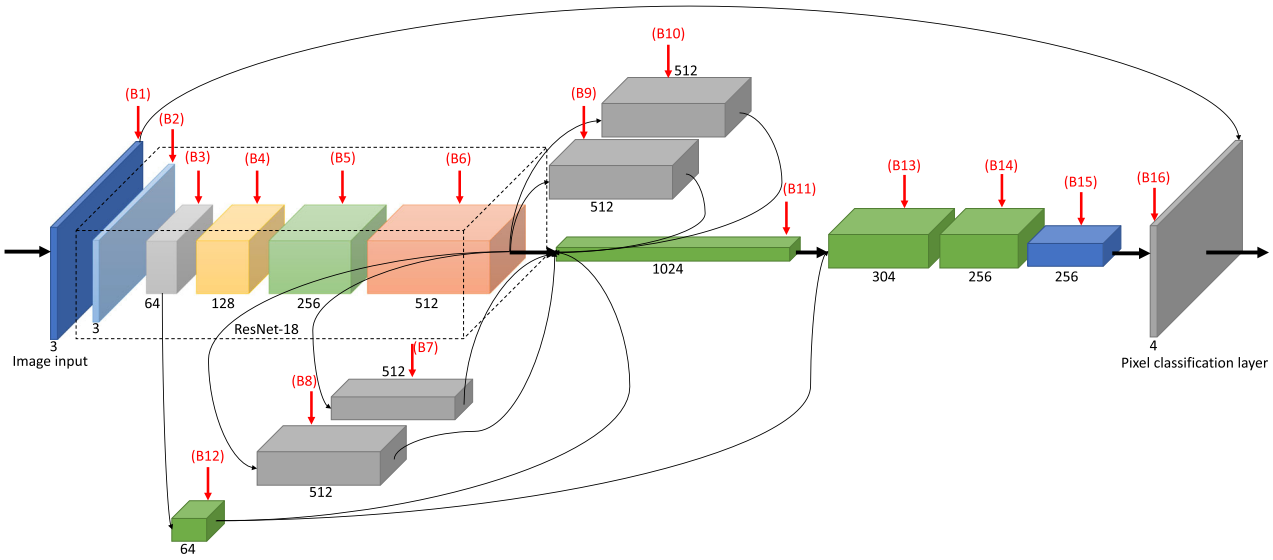

FIGURE 5 Convolutional neural network architecture implemented in this study based on Deeplab v3+ ResNet-18 [Color figure can be viewed at wileyonlinelibrary.com]

were modified and fine-tuned. They are the directed acyclic graph (DAG) network: (i) Deeplab v3+ ResNet-18, which is abbreviated as ResNet-18 in the following content (Chen et al., 2017, 2018); SegNet: (ii) VGG-16, and (iii) VGG-19 (Simonyan & Zisserman, 2014; Table 1). ResNet was developed by He et al. (2016) and extensively uses the batch normalization layers to accelerate the network training but lacks fully connected layers at the end of the architecture. VGG networks, in contrast, are very deep but are time and memory consuming. All three networks require a minimum image size of 224-by-224 pixels. ResNet-18 was trained with original and resized images, while VGGs were trained with only resized images using the GPU-based platform. To better understand the computational characteristics of the networks, Figure 5 visualizes the overall architecture of ResNet-18 with 101 layers after Deeplab v3 is added and the activation channels of the convolutional layers used. The architecture can be divided into 16 processing blocks: B1–B16. The architectures of VGG-16 and VGG-19 SegNet (Table 1) are omitted because they can be found in some other studies (Majeed et al., 2020). The specific units in the modified Deeplab v3+ ResNet-18 architecture are as follows, where ResNet-18 functioned as the encoder and Deeplab v3 functioned as the decoder:

i. Block 1 (B1): First, preprocessed foreground images (1080 × 1920 × 3 or 540 × 960 × 3) are loaded to feed to the network.
ii. Block 2–6 (B2–B6): The images are processed in the original ResNet-18 blocks, and these contain a series of convolutional

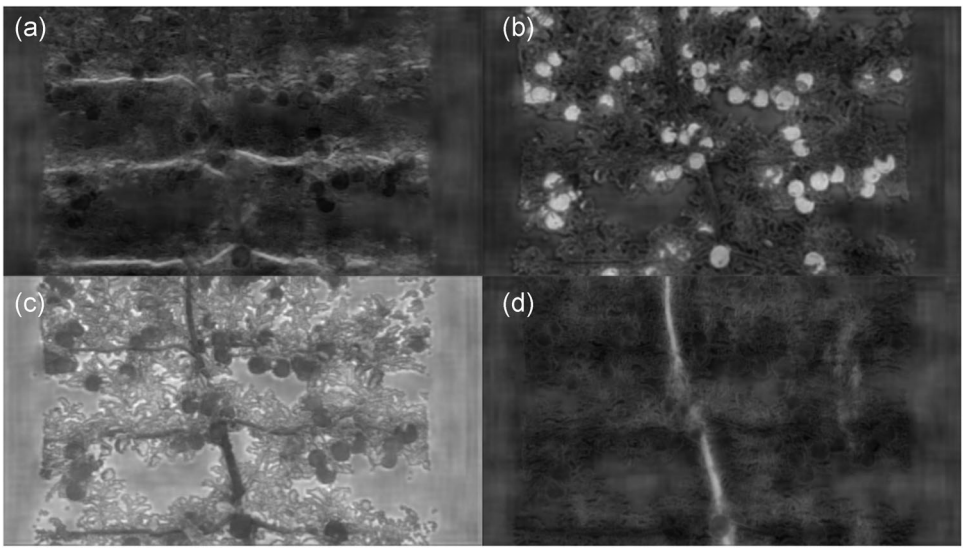

FIGURE 6 Positive activation channels for four classes; (a) branches; (b) apples; (c) leaves; and (d) trunks at the scorer convolutional layer of the modified Deeplab v3+ ResNet-18

TABLE 2 Some of the major parameters used in training the networks; ResNet-18, VGG-16, and VGG-19

Parameter	Deeplab v3+ ResNet-18	VGG-16	VGG-19
Optimization algorithm	SGDM	SGDM	SGDM
Initial learn rate	1×10^{-2}	1×10^{-2}	1×10^{-2}
Learn rate drop period	10	–	–
Learn rate drop factor	0.3	–	–
L_2 regularization	1×10^{-4}	1×10^{-4}	1×10^{-4}
Gradient threshold	–	0.07	0.07
Mini-batch size	8	1	1

Abbreviation: SGDM, Stochastic gradient descent with momentum.

layers, batch normalization layers, rectified linear unit (ReLU) layers, and max-pooling layers. Convolutional layers are the core building blocks of CNNs where the layer parameters consist of a set of learnable filters. These blocks automatically compute the output of locally connected neurons to regions from the input. After each convolutional layer, there is one batch normalization layer and/or one ReLU layer connected. The ReLU layer simply thresholds the negative activations at zero and only passes the positive activations (further explained in Section 4) to the next layer that vastly accelerates the convergence of optimization algorithms, such as stochastic gradient descent with momentum (SGDM). During the early stages, the network starts learning some shallow features such as the edges and the colors/shapes. The network always tends to learn more features from those positive activations throughout the entire training process because of the ReLU layers. In this combination of Deeplab v3+ ResNet-18, the last ResNet-18 block employs atrous convolutions, a tool to adjust field-of-view of the filters, with various dilation rates. It adopts atrous spatial pyramid pooling and bilinear up-sampling for the decoder (Deeplab v3) based on the ResNet-18 architecture as the main feature extractors (Chen et al., 2017). The network learns some abstraction features as the layers go deeper, and these are often extremely difficult for a human to distinguish.

iii. Block 7–10 (B7–B10): After the original ResNet-18, four blocks are connected in parallel to process the feed-in image data. Each block contains a convolutional layer with 512 activation channels, a batch normalization layer, and a ReLU layer.

iv. Block 11–15 (B11–B15): Next, there are a series of blocks, and each of which is followed by a convolutional layer with different activation channels: 1024, 64, 304, and 256, respectively, a batch normalization layer, and a ReLU layer. B15 is an exception. It only contains a convolutional layer of a scorer with 256 activation channels and a transposed convolutional layer to up-scale the sample images. All positive activation channels for four classes of branches (Figure 6a), apples (Figure 6b), leaves (Figure 6c), and trunks (Figure 6d) are displayed together in Figure 6. In these instances, the brighter parts are the positive activations, whereas the darker parts are the negative activations. These results confirm that the modified ResNet-18 is working effectively to segment out all classes of interest by automatically learning their features.

v. Block 16 (B16): Finally, the last block contains a center crop layer, a softmax layer "softmax-out," and a pixel classification layer labels with four classes (i.e., branches, apples, leaves, and trunks) to generate an output image with learned semantic segmentation results. The crop layer takes two bottom layers, including input and convolutional layers, and output as a single layer to match the output image size to the input image size. Besides, the softmax layer is placed right before the output layer to map the nonnormalized output to a probability distribution of the predicted output classes.

2.4.2 | Network training, validation, and testing

The full data set with 674 images of medium-density foliage canopies of Fuji was randomly partitioned into three parts: 70% images (472) for training, 15% (101) for validation, and 15% (101) for testing. The performance of the trained networks was further assessed on other image datasets, including 15 images of light-density foliage canopies of Pink Lady and 58 images of Envy and 38 images of Scifresh considered as high-density foliage canopies. The employed networks were fine-tuned individually and repeatedly using stochastic gradient descent with momentum (SGDM; Equation 1) as the optimization (backpropagation learning) algorithm (Murphy, 2012) for all three networks. The training process was completed when the validation accuracy converged. Some critical parameters defining the network training process are listed in Table 2. The initial learning rate determines the speed of the training process. In this study, the learning rate was configured to drop by a drop factor after each interval of 10 epochs. L_2 regularization refers to weight decay that helps reduce the chances of network overfitting (Equations 2 and 3). Mini-batch size is the subset of image data used at each iteration, and gradient threshold was used to stabilize the training process when a higher learning rate was employed.

Image augmentation was another technique used in improving the training process. Image data were augmented during the training stage to increase the training samples provided to the networks. The augmentation technique applied was right/left reflection and x/y-axis translation by ±5 pixels. Barth et al. (2018) provided more information associated with data synthesis/augmentation methods.

$$\theta_{\ell+1} = \theta_{\ell} - \alpha \nabla E(\theta_{\ell}) + \gamma(\theta_{\ell} - \theta_{\ell-1}). \quad (1)$$

where θ refers to the parameter vector, ℓ refers to iteration number, α refers to the learning rate ($\alpha > 0$), $E(\theta)$ refers to the loss function, $\nabla E(\theta)$ refers to the gradient of the loss function, and γ determines the contribution of the previous gradient step to the current iteration.

$$E_R(\theta) = E(\theta) + \lambda\Omega(w), \tag{2}$$

$$\Omega(w) = \frac{1}{2}w^T w, \tag{3}$$

where E_R refers to the regularization loss, λ refers to the regularization coefficient, and w refers to the weight vector.

2.4.3 | Network evaluation

Once the network was completely trained and validated, the performance of the network on the test data set was evaluated using region-based measures. This included normalized confusion matrix (**C**), per-class accuracy (PcA), per-image/mean intersection over union (IoU), and the contour-based measure of per-image/mean BFScore (Csurka et al., 2013). The normalized confusion matrix was another measure used that is more revealing than a regular confusion matrix when the number of pixels in each class is imbalanced. For example, in this case, the number of pixels of background is much greater than of the trunk, branches, and apples. PcA measures the proportion of correctly classified pixels for each class and provides the average value over all classes based on the normalized confusion matrix. This measure gives general information on the accuracy of the prediction. However, it has significant drawbacks for the data set with a large background class, for example, leaves in this study. The background class could absorb false predictions with no influence on other object class accuracy.

IoU has been recognized as one of the more efficient measures for assessing semantic segmentation performance. IoU measures the intersection over the union between predicted classes and the ground-truth labels for each class and averages the results (Equations 4–6; Csurka et al., 2013). Both per-image and mean IoU results are reported in this study.

$$\text{IoU} = \frac{\sum_{i=1}^{N} \frac{C_{ii}}{G_i + P_i - C_{ii}}}{N}, \tag{4}$$

$$G_i = \sum_{j=1}^{N} C_{ij}, \tag{5}$$

$$P_j = \sum_{i} C_{ij}, \tag{6}$$

where N refers to the number of classes, $\boldsymbol{C}$ refers to the pixel-level confusion matrix as discussed, $\boldsymbol{C_{ii}}$ refers to the number of pixels with

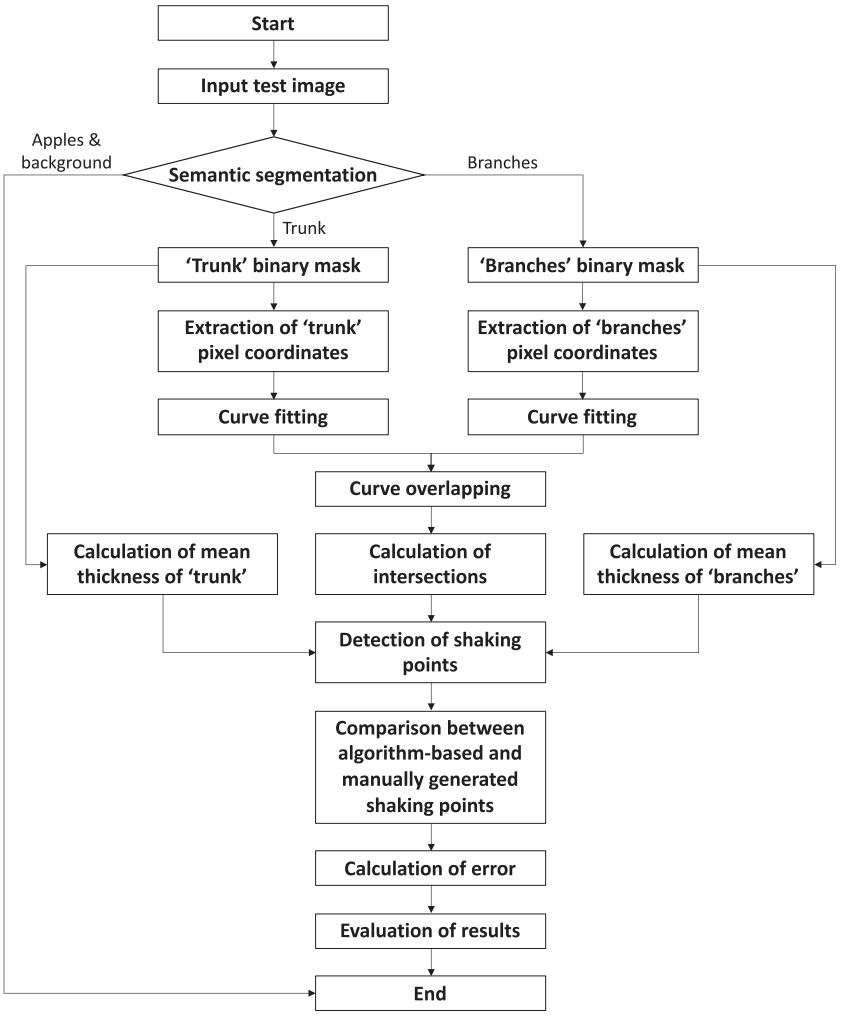

FIGURE 7 Flow chart of the shaking points detection technique using the segmented pixels of branches and trunks

both ground-truth label and prediction label being i, C_{ij} refers to the number of pixels with ground-truth label i but whose prediction label is j, G_i refers to the total number of pixels labeled with i, and P_i and P_j refer to the total number of pixels predicted as i and j, respectively. IoU was also weighted (weighted IoU) by the number of pixels in respective classes.

Although IoU provides a comparatively more representative measure in assessing the performance of the semantic segmentation models, it is limited in terms of representing class boundaries. The contour-based measure of mean BFScore is used widely in evaluating class boundaries between the ground-truth and predicted classes in semantic segmentation. Precision (P^c) and recall (R^c) are used in estimating BFScore (Equations 7–9; Csurka et al., 2013). Both per-image and mean BFScore results are reported.

$$F_1^c = \frac{2 \bullet P^c \bullet R^c}{R^c + P^c}, \tag{7}$$

$$P^c = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{8}$$

$$R^c = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{9}$$

Here, c refers to a class, TP refers to true positives, FP refers to false-positives, and FN refers to false-negatives.

2.5 | Estimating shaking locations

Once the target classes of branches and trunks were successfully segmented and identified, suitable shaking locations were estimated on those branches (Figure 7). Shaking locations were estimated based on the optimal locations suggested by He et al. (2019), who found that shaking at the branch bases (right next to the trunks) was more effective in removing fruits than shaking at the middle of the branches. The significant steps, in this process, included:

(i) Binary masks of branches and trunks (1920 × 1080 pixels) on canopy images were obtained based on the results generated by trained CNNs. A morphological operation was performed to remove the objects containing fewer than 600 pixels for both classes. The rest of all pixel coordinates of object masks were extracted for fitting polynomial curves for branches (Equation 10) and trunks (Equation 11). The performance of curve fitting was assessed using R^2. When all curves were mapped together, the intersections of curves (x_i, y_i) were calculated:

$$f(x) = p_n x^n + p_{n-1} x^{n-1} + \cdots + p_2 x^2 + p_1 x + p_0, \tag{10}$$

$$f(y) = q_n y^n + q_{n-1} y^{n-1} + \cdots + q_2 y^2 + q_1 y + q_0, \tag{11}$$

where x represents the pixel coordinates along the x-axis in an image, y represents the same in the y-axis, n represents the degree of a polynomial, and p and q are the coefficients of the polynomial (real numbers).

(ii) Mean thicknesses of trunks ($\dot{d_t}$, along the x-axis) and branches ($\dot{d_b}$, along the y-axis) in terms of the number of pixels were calculated based on masks (Equations 12 and 13).

$$\dot{d_t} = \frac{1}{y}\sum_{i=1}^{y} d_x, \tag{12}$$

$$\dot{d_b} = \frac{1}{x}\sum_{i=1}^{x} d_y, \tag{13}$$

where t refers to trunks, b refers to branches, $\dot{d_t}$ is used for detecting the base shaking points by estimating the nearest branches locations to trunks (x_a, y_a), $\dot{d_b}$ is used for calculating the error tolerance of the detected shaking points along the y-axis (y_{error}; solved in Equation 14):

$$y_{\mathrm{error}} = \pm\frac{\dot{d_b}}{2}. \tag{14}$$

(iii) Shaking points (x_m, y_m) were selected manually and were compared with the points selected by the algorithm, with an assumption of $x_a = x_m$. With expertise and experience in operating shake-and-catch apple harvester, the authors of this study subjectively selected the suitable shaking points near branch bases using the segmented images, including tree trunks and branches. The selection criterion is simple: finding the nearest point (not occluded by leaves and fruit) on each segmented branch to the segmented trunks based on the semantic segmentation results. The position difference on the y-axis can thus be calculated (Equation 15), and the error tolerance can be determined by solving Equation (14). Finally, the performance of algorithm-based shaking point selection is reported as "good" or "poor" according to Equation (16). In total, 20 test images of Fuji canopies were randomly selected for evaluation purposes.

TABLE 3 Training and validation results with ResNet-18, VGG-16, and VGG-19

Results	Deeplab v3+ ResNet-18		VGG-16	VGG-19
Image size (pixel)	1080 × 1920	540 × 960	540 × 960	540 × 960
Validation accuracy[a] (%)	96.00	94.74	93.39	94.11
Validation loss	0.08	0.11	0.14	0.13
Elapsed time (s)	57,050.67	6,912.00	12,347.93	17,589.99

[a]Accuracy refers to overall per-class accuracy (PcA).

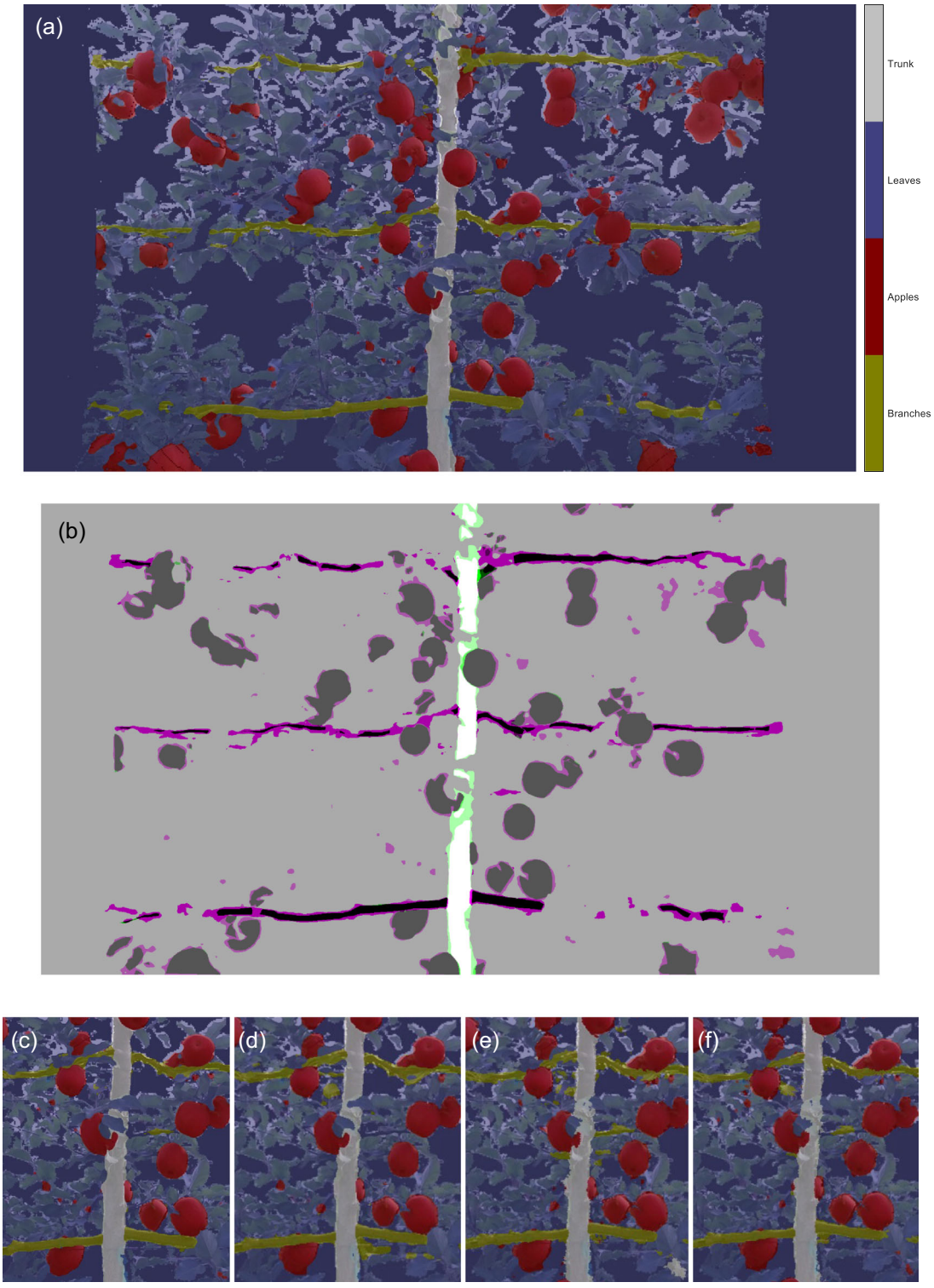

FIGURE 8 (a) An example of test results of semantic segmentation using Deeplab v3+ ResNet-18 with the original image size; (b) comparison of test results and ground-truth data (magenta and green regions highlight the areas where the test result varied from the ground-truth; (c) and (d) zoomed-in local boundaries resulted by Deeplab v3+ ResNet-18 with the original and reduced image sizes, respectively; (e) the same with VGG-16; and (f) the same with VGG-19 [Color figure can be viewed at wileyonlinelibrary.com]

$$y_d = |y_a - y_m|, \tag{15}$$

$$\begin{Bmatrix} y_d \le y_{\text{error}}, \\ y_d > y_{\text{error}}. \end{Bmatrix} \tag{16}$$

Here, m refers to manual selection, and d represents the difference between algorithm-based and manual selections along the y-axis. One shaking point was detected for each branch. The number of pixels was used as a measurement unit during the evaluation process.

3 | RESULTS AND DISCUSSION

3.1 | Training and validation on the Fuji cultivar data set

Images of medium foliage density apple canopies of Fuji cultivar were used to train and validate three CNNs. ResNet-18 achieved a higher validation PcA of ~95% with a lower loss value of 0.11 using 540 × 960 image size (Table 3). Comparatively, both VGG-16 and VGG-19 were found to achieve slightly lower validation accuracies of 93%–94% and greater loss values of 0.13–0.14 with the same set of image sizes. In terms of computational time, only about half and one-third of the time was consumed by ResNet-18 (6912 s) for training and validation compared to the other two networks on a single GPU. This was mainly because of its DAG network architecture, and it lacked the fully connected layers (Chen et al., 2018), potentially slowing down the processing speed of the networks because every input is connected to every output by specific weights. Besides, two different input image sizes of 1080 × 1920 versus 540 × 960 were used, and the performance was compared using ResNet-18 (Table 3). The results revealed that a higher accuracy could be achieved using higher resolution images with 96% of validation accuracy and 0.08 loss value. However, it also took about eight times longer to finish the entire process.

3.2 | Testing on Fuji cultivar data set

The trained networks were then tested on the remaining 15% of the Fuji cultivar data set. Figure 8 visualizes the results of a test image, which is successfully segmented into four target classes: branches in yellow, apples in red, leaves in blue, and trunks in white (Figure 8a). The results show, as expected, that ResNet-18 with an original image size performed the best in terms of segmenting the images as well as preserving the boundary information of objects (Figure 8c). Zoomed-in views are presented in Figure 8c–f and show that ResNet-18 performed better than VGG-16 and VGG-19 regarding the smoothness of the boundary information, especially with trunks and branches boundaries on resized images. Test results were then compared against the ground-truth data (Figure 8b), where the misclassified pixel-regions (false-positives) were highlighted in both magenta and green colors. The results showed that most misclassified regions were found with VGGs with reduced size pixel resolution, particularly the regions closer to the tree branches, whereas ResNet-18

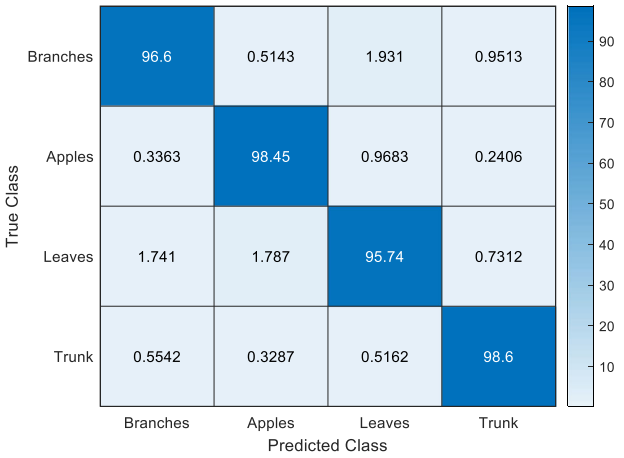

FIGURE 9 Normalized confusion matrix (%) comprising the pixels in the true class in the vertical-axis and the predicted class in the horizontal axis based on the semantic segmentation results generated by the modified Deeplab v3+ ResNet-18 model. The matrix was generated using images with original pixel resolution [Color figure can be viewed at wileyonlinelibrary.com]

performed the best when images with original pixel resolution were used. With ResNet-18, most of the image pixels were accurately classified into the corresponding classes (Figure 9), leading to ~99% of trunks pixels correctly predicted as true class trunks, followed by 98% for apples. True prediction for branches was slightly lower than that for trunks and apples, which might be because of a lower percentage of pixels belonging to branches (1.15%) than to trunks (1.25%) and apples (6.20%) in the images. In addition, a more distinct color (redness) and shape (roundness) were associated with the apples. Two of the most common misclassifications were found between branches and leaves because of the similarities of class features, such as color and texture.

As discussed in Section 2, IoU and BFScore were used, besides PcA, to improve the insights in network performance. Mean IoU and BFScore per image obtained with the three CNNs on images with two different pixel resolutions are presented in Figure 10. ResNet-18, again, achieved the best results on both using full resolution images. For all images, the mean IoU per image was found to be 0.62 or higher (Figure 10a), and the mean BFScore per image was found to be 0.80 or higher (Figure 10b).

In contrast, all three CNNs achieved relatively lower IoU and BFScore with downsampled images. For example, ResNet-18 achieved an IoU of 0.62 or more for about 76% of the test images (77 out of 101). The results also showed that ResNet-18 performed substantially better with original and reduced resolution images than VGGs to reproduce the overlapped areas between prediction and ground-truth data, where VGG-16 had the worst performance. In terms of BFScore, about 88% (89 images) were found to have 0.80 or higher mean BFScore with ResNet-18, which was slightly better than the same achieved with VGGs. The results indicated that ResNet-18 is better in preserving the boundary information of objects (Figure 8c–f) with either image size, followed by VGG-19.

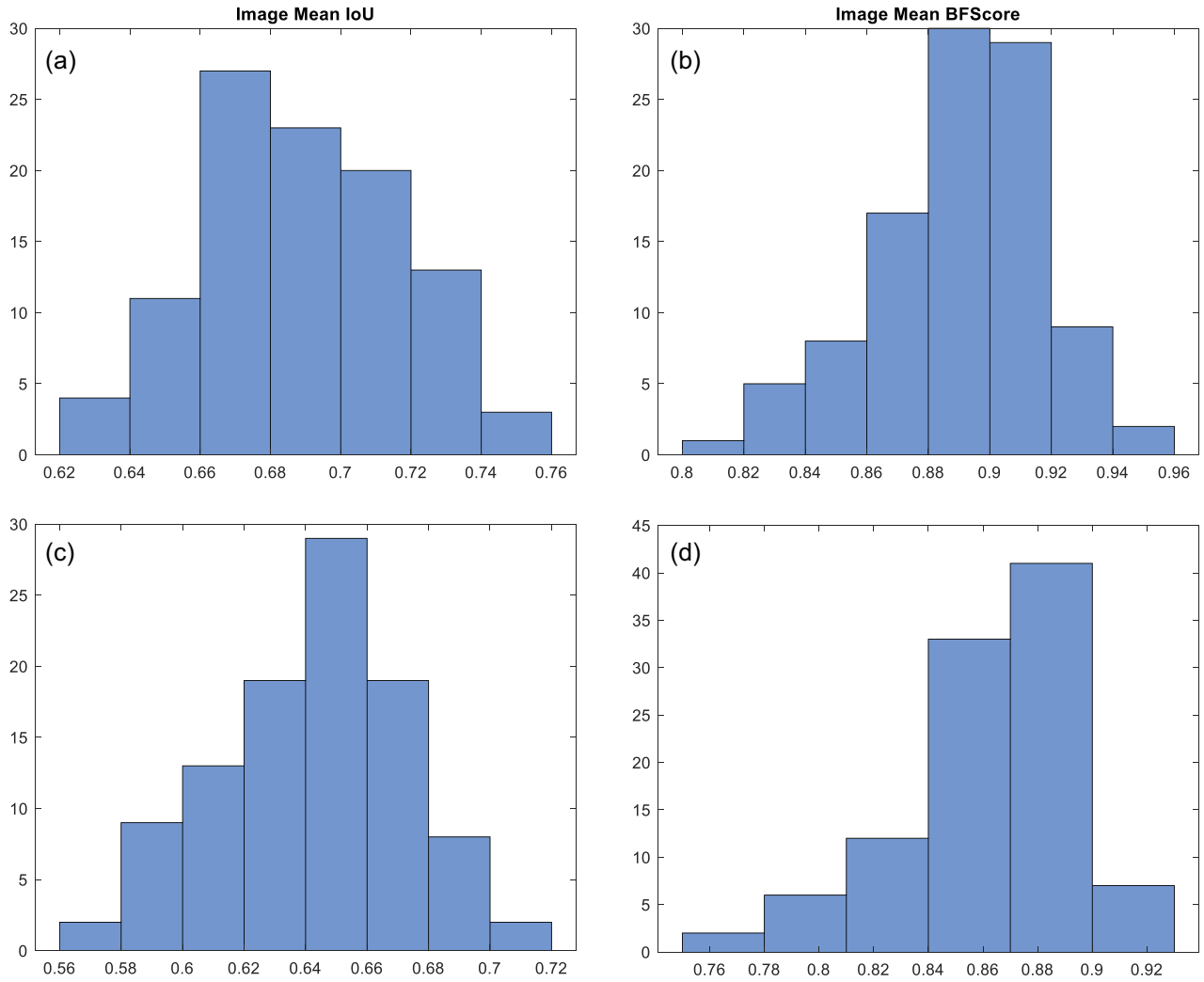

FIGURE 10 Histograms of mean IoU and mean boundary-F1 score (BFScore) using Deeplab v3+ ResNet-18 with original image size (a and b) and with resized images (c and d). In these plots, the *y*-axis represents the total number of images. IoU, intersection over union [Color figure can be viewed at wileyonlinelibrary.com]

In addition to the per-image results discussed, per-class results were also compared (Table 4). Overall, ResNet-18 with full image size achieved the best results with a mean PcA of 97%, mean IoU of 0.69, and mean BFScore of 0.89, followed by the same network on the resized images with a mean PcA of 97%, mean IoU of 0.64, and mean BFScore of 0.86, and then VGGs with a mean PcA of 96%, mean IoU of 0.61–0.62, and mean BFScore of 0.81–0.84. IoU results varied substantially among four classes; IoU was 0.96 for leaves, while it was 0.40 for branches with ResNet-18 being performed on the original images. IoU is calculated using both false-positives and true-negatives for each class; therefore, classes with a greater number of pixels (leaves in this study) can have better IoU than a class with a lower number of pixels (branches and trunks). This variation was also noticed by Zabawa et al. (2019) when they segmented individual grapes for early yield estimation. Moreover, a 0.40 IoU for branches was considered acceptable because there would be an average overlapping area of ~57% when IoU is 0.4 based on Equations (4)–(6). In the research conducted by Zhang et al. (2018), an IoU of 0.3 was considered positive and acceptable. For the trunks class, an IoU of 0.63 referred to an average overlapping area of ~77% (Figure 11). In terms of BFScore, similar trends could be found with lower values for branches and trunks of 0.82–0.89 and higher values for apples of 0.93 when using ResNet-18 on the original images, indicating that apples preserved slightly better local boundary information than other objects probably because of its distinct color and shape features.

When the best performing model (ResNet-18 with original image resolution) and the worst performing model (VGG-16 with reduced image resolution) were compared on Fuji canopy images, the semantic segmentation results of branches and trunks are remarkably different. However, the semantic segmentation results for apples and leaves are only marginally different. For example, the IoUs of branches and trunks increased from 0.27 to 0.40 (by 48%) and from 0.54 to 0.63 (by 17%), while the IoUs of apples and leaves increased only from 0.70 to 0.78 (by 11%) and from 0.93 to 0.96 (by 3%), respectively. This improvement was highly critical for accurately identifying

TABLE 4 Network evaluations in terms of per-class accuracy (PcA), intersection over union (IoU), and boundary F1-score (BFScore)

Evaluation Measure	PcA (%)				IoU				BFScore			
Network	Deeplab v3+ ResNet-18		VGG-16	VGG-19	Deeplab v3+ ResNet-18		VGG-16	VGG-19	Deeplab v3+ ResNet-18		VGG-16	VGG-19
Image size[a]	Full	Reduced	Reduced	Reduced	Full	Reduced	Reduced	Reduced	Full	Reduced	Reduced	Reduced
Branches	96.60	95.54	94.55	94.62	0.40	0.30	0.27	0.30	0.82	0.75	0.69	0.74
Apples	98.46	97.59	97.68	97.46	0.78	0.76	0.70	0.71	0.93	0.93	0.89	0.91
Leaves	95.74	94.45	93.06	93.72	0.96	0.94	0.93	0.94	0.92	0.90	0.87	0.89
Trunk	98.60	98.47	97.16	97.96	0.63	0.58	0.54	0.54	0.89	0.87	0.81	0.83
Mean	97.35	96.51	95.61	95.94	0.69	0.64	0.61	0.62	0.89	0.86	0.81	0.84
Weighted	–	–	–	–	0.94	0.92	0.90	0.91	–	–	–	–
Computational speed[b] per image (s)	1.29 ± 0.10a	0.35 ± 0.05c	0.44 ± 0.04b	0.47 ± 0.03b	–	–	–	–	–	–	–	–

[a]Full and reduced image sizes referred to 1080 × 1920 and 540 × 960 pixels, respectively.
[b]ComputatioOnal speed was calculated based on randomly tested 10 images (mean ± *SD*) for each network; different letters refer to a statistically significant difference between the columns using Tukey honestly significant difference (HSD) test ($p < .05$).

FIGURE 11 Example of a segmented trunk in red and branches in yellow mapped onto its foreground image [Color figure can be viewed at wileyonlinelibrary.com]

the tree trunks and branches, providing a basis for automating shake-and-catch harvesting for apples. In addition, good semantic segmentation accuracy for apples also helps improve the harvesting efficiency by targeting the specific shaking areas, for example, to avoid hitting the apples. In terms of computational speed, ResNet-18 (0.35 s per image) was significantly faster than VGGs (0.44–0.47 s) with the resized images when tested using the Tukey honestly significant difference (HSD) test ($p < .05$). Although the computational time was increased to 1.29 s when the higher resolution images were used in testing the network (Table 4), the performance was considered acceptable for near-real-time application in automated, shake-and-catch harvesting.

3.3 | Network testing with image data sets from different crop cultivars

ResNet-18, which outperformed other networks, was adopted for further analysis with the data set collected from different crop cultivars than those used in earlier training and testing with varying foliage density, demonstrating the robustness of the algorithm used. Three new image datasets used for this extended testing were collected from orchards with lighter foliage density apple cultivars (Pink Lady; Figure 12a) and higher foliage density cultivars (Envy, Figure 12b, and Scifresh, Figure 12c). Qualitatively, good trajectories of trunks and branches were predicted with the new data sets, even when the branches were heavily occluded by leaves or apples, as illustrated in Figures 12b,c. The quantitative results are presented in Table 5. These results show that the ResNet-18-based model performed generally well on images from different apple cultivars with varying foliage densities that were never presented to the network during the training process. As expected, the best results were found in canopies with light foliage density of Pink Lady with a mean PcA of 96%, mean IoU of 0.75, and mean BFScore of 0.92 while the IoUs for branches and trunks reached 0.47 and 0.72, respectively. These results were slightly better than the test results with the original data set of Fuji canopies (Table 4). The improvement could be attributed

FIGURE 12 Examples of segmented trunks in red and branches in yellow mapped onto corresponding foreground images of light-density Pink Lady canopies (a); and high-density canopies of Envy (b) and Scifresh (c). The semantic segmentation results were generated by the Deeplab v3+ ResNet-18 model with original image resolution [Color figure can be viewed at wileyonlinelibrary.com]

to fewer occlusions to branches because of relatively lighter foliage density.

The trained network also achieved relatively good performances on canopies with higher foliage density, especially with the Scifresh cultivar (Figure 12c). For example, IoUs of 0.41 and 0.62 were achieved for branches and trunks, indicating satisfactory predictions of branch and trunk trajectories as the trained network provided 58% and 77% of overlapping areas between the predicted and ground-truth regions, respectively. These results were similar to what was achieved with the medium foliage density canopies of Fuji cultivar originally tested in this study. However, the results show that the network performs relatively poorly on the Envy data set, which represents one of the highest foliage density canopies (Figure 12b). On this cultivar, IoU achieved for branches and trunks was 0.34, with 51% of the overlapping area and 0.56 with 72% of overlapping area, respectively. Similarly, the BFScore achieved were 0.65 and 0.80 for

TABLE 5 Evaluations of network performance on canopy datasets with varying foliage density in terms of per-class accuracy (PcA), intersection over union (IoU), and boundary F1-score (BFScore)

Evaluation measure	PcA (%)			IoU			BFScore		
Canopy type	Light	High		Light	High		Light	High	
Cultivar	Pink Lady	Envy	Scifresh	Pink Lady	Envy	Scifresh	Pink Lady	Envy	Scifresh
Branches	91.55	90.06	77.85	0.47	0.34	0.41	0.85	0.65	0.71
Apples	95.72	98.27	96.70	0.84	0.81	0.76	0.96	0.92	0.91
Leaves	96.14	96.25	96.35	0.96	0.96	0.96	0.95	0.90	0.92
Trunk	98.61	97.15	86.91	0.72	0.56	0.62	0.93	0.80	0.86
Mean	95.50	95.43	89.45	0.75	0.67	0.69	0.92	0.82	0.85
Weighted	–	–	–	0.93	0.94	0.93	–	–	–
Computational speed[a] per image (s)	1.24 ± 0.02a	1.25 ± 0.02a	1.24 ± 0.02a	–	–	–	–	–	–

Note: The network used was Deeplab v3+ ResNet-18 and the input images were of original/higher resolution

[a]Computational speed was calculated based on randomly tested 10 images (average ± SD) for each network; different letters refer to a statistically significant difference between the columns using Tukey honestly significant difference (HSD) test ($p < .05$).

TABLE 6 Comparing the order/degree (n) of polynomials in terms of R^2 in fitting branches and trunks

R^2 of Polynomials				
Degree	***n* = 2**	***n* = 3**	***n* = 4**	***n* = 5**
Branches	0.33 ±0.21b[a]	**0.40 ±0.20ab**	0.46 ±0.20ab	0.48 ±0.20a
Trunks	0.63 ±0.27a	**0.67 ±0.25a**	0.68 ±0.25a	0.69 ±0.24a

[a]Mean ± *SD* over 10 randomly selected test images; different letters (a and b) refer to a statistically significant difference between the columns using Tukey honestly significant difference (HSD) tests ($p < .05$).

branches and trunks, respectively. The obtained IoUs could still be acceptable, as illustrated in Figure 8b. Qualitatively, most of the areas of branches were successfully covered by the predictions with acceptably precise boundary descriptions. Overall, about 1.24–1.25 s per image was taken by the network to process one image, and this time was insignificantly different for all levels of canopies from Pink Lady to Scifresh (Figures 11 and 12b,c; Table 5) using the Tukey HSD test ($p < .05$). To further increase the identification accuracy of the networks, some higher resolution images might be used, but this could also reduce the computational speed at the same time. To address the issue, large images could be divided into smaller sections before feeding to the networks to increase the identification accuracy without sacrificing the computational speed much (e.g., Zabawa et al., 2019).

Although it is important to test the implementation of the trained networks on datasets never previously seen by the network for demonstrating the robustness of the model, only about 20% of the published studies adopted network test measures testing outside of the data set used in model training (Kamilaris & Prenafeta-Boldú, 2018). Recently, two other studies conducted multiclass object detection on SNAP fruit tree architecture during foliage season using Faster R-CNN (Gao et al., 2020; Zhang, Karkee, et al., 2020). The work from Zhang, He, et al. (2020) achieved 0.45 s per image computational speed with the network only but used a total of 3.14 s for the entire detection process. Their model was tested with relatively lower resolutions images (360 × 640) and also was not generalized for practical adoption through the testing of different cultivars. Overall, the results show that the model used in our study (modified,

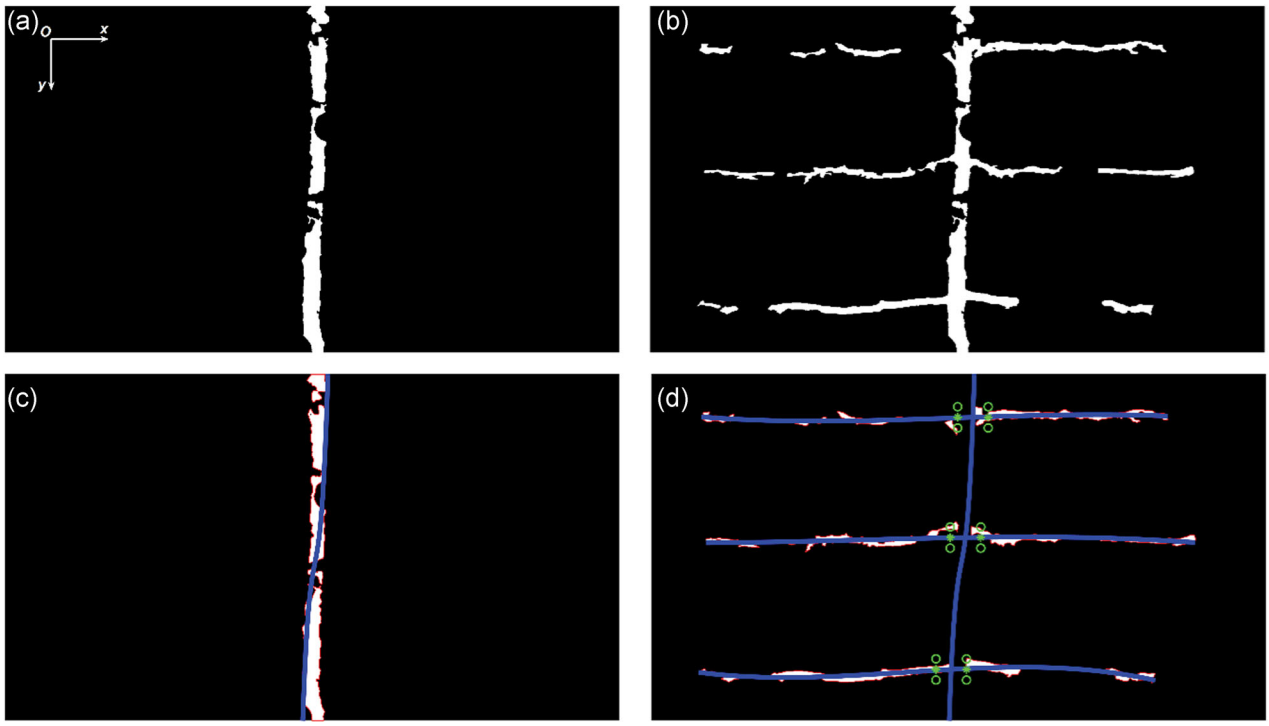

FIGURE 13 Illustrations of shaking points selection process described in Figure 7: binary mask of tree trunks (a); binary mask of tree branches (b); fitted polynomial curve with degree $n = 3$ in a blue vertical line over trunks (c); and fitted and mapped polynomial curves with degree $n = 3$ in blue horizontal lines over branches (d). In the plots, green "*" represents the estimated shaking points at branch bases derived by solving Equations (10)–(12), while green "o" represents the error tolerance for the points along the y-axis solved in Equation (14) [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 7 Evaluation of shaking point estimation algorithm against manually selected shaking points

Trunks thickness[a] ($\dot{d_t}$)	Branches thickness[a] ($\dot{d_b}$)	Error tolerance[a] (y_{error})	Error[a] (y_d)	
			Good	Poor
111.31 ± 33.19[b]	55.52 ± 6.14	27.76 ± 3.07	10.96 ± 7.34	42.63 ± 12.91
Overall percentage (%)			71.67	28.33

[a]All units are in pixels with an image resolution of 1920 × 1080.
[b]Mean ± *SD* over 20 randomly selected test images where six shaking points were evaluated per image.

fine-tuned ResNet-18) could generally be used as a robust and generic model to segment out canopy images from varying cropping systems and environmental conditions, including crop canopies with light to medium/high foliage densities. The three foliage density levels tested in this study primarily represent the overall canopy conditions of SNAP trellis-trained tree architectures in the Pacific Northwest region of the United States. Therefore, the annotated image data set in this study could be further utilized to either train other potential CNNs or reproduce the results with the same networks discussed for any other branch, trunk, or apple identification tasks in the agricultural field (WSU Research Exchange URI: http://hdl.handle.net/2376/17529).

3.4 | Estimation of shaking locations

A curve fitting algorithm was developed to detect the shaking locations near branch bases, as illustrated in Figure 7. Polynomial curves are often adopted to represent irregular line curves (Zhang et al., 2018) and were used in this study to represent trunks and branches. To optimize the degrees (n) of the polynomials, 10 test images were randomly selected to assess the performance of polynomials of varying degrees in representing the trunks and branches (Table 6). The results showed that the R^2 value of the fitted curve increased with increasing polynomial complexity. However, the results show that tree trunks were often overfitted with polynomials with 4th or 5th degrees because of the small, scattered object masks caused by false-positive pixels. Therefore, a 3rd-degree polynomial was adopted for fitting the trunks and branches in this study. With such a polynomial, the averaged R^2 achieved was 0.40 for branches and 0.67 for trunks. Figure 13 shows the steps (Figure 13a,b) and results (Figure 13c,d). The polynomial curves were fitted for trunks with a blue vertical line (Figure 13c) and branches with three blue horizontal lines (Figure 13d). The algorithm-based shaking points at each branch base were detected and visualized using green "*" symbols in Figure 13d. In addition, the error tolerance of detections along the y-axis was visualized in the same figure using green "o" symbols.

The curve fitting and estimation of shaking points were performed using 20 randomly selected images, leading to 120 shaking points for evaluation purposes. Manually selected shaking points were generated as ground-truth data to evaluate the performance of algorithm-based selections using images at a 1920 × 1080 resolution (Table 7). As per Equations (14)–(16), the mean error tolerance along the y-axis (y_{error}) between the estimated and manually selected shaking points was approximately 27.8 pixels. The results indicate that about 71.7% of selected points were considered "good," corresponding to the definition (Equation 16) where the mean error along the y-axis (y_d) was ~11.0 pixels. The rest of 28.3% of the points had "poor" localization performances with a relatively high error of 42.6 pixels on average. It is noted that error tolerance could be increased for automated shake-and-catch harvesting by adopting a wider grip in the shaking end-effector, thus potentially avoiding or minimizing the impact of "poor" performance in shaking point localization.

Lastly, the overall computational time was calculated for the entire process of tree branches/trunks identification, curve fitting, and shaking points selection. The results show that the curve fitting and shaking point selection (~1.3 s) took about the same time as CNNs-based semantic segmentation (~1.4 s) on average; approximately 2.7 s was needed in total per image. Based on this results, it will take approximately 0.5 s per shaking point for image processing, curve fitting, and shaking point selection, and this should be practically applicable for near-real-time application in automated shake-and-catch harvesting as each shaking actuation cycle would take at least 2–5 s (He et al., 2019).

4 | CONCLUSIONS

In this study, a complete pipeline work was first provided to identify tree branches and trunks in canopies with varying foliage densities trained to SNAP tree architectures for the automated mass harvesting of apples. Machine vision system under natural field environment and CNNs-based deep learning techniques of semantic segmentation were employed. Four different pixel classes were defined as branches, trunks, apples, and leaves (background). A total of 674 images were acquired from a commercial Fuji orchard with medium foliage density canopies. With a full pixel resolution of 1080 × 1920 and a reduced pixel resolution of 540 × 960, these images were used to train, validate, and test three different CNNs Deeplab v3+ ResNet-18, VGG-16, and VGG-19, that were modified and fine-tuned for this study. Moreover, to test the robustness of the trained network with ResNet-18, a new set of images was collected in tree canopies with varying foliage densities offered by Pink Lady, Envy, and Scifresh cultivars. The performance of these networks in image semantic segmentation was assessed and compared using PcA, IoU, and BFScore measurements on all test datasets. Finally, a curve fitting technique was used to model tree trunks/branches and to

estimate shaking points on those branches for automated shake-and-catch harvesting. The estimated shaking points were compared against manually selected points on the same images. Specific conclusions from this study are presented as follows:

- ResNet-18 using full image resolution performed the best among three CNNs tested in this study with a mean PcA of 97%, mean IoU of 0.69, and mean BFScore of 0.89 per image basis on images collected in a V-trellised Fuji apple orchard. The network performance on per-class basis was also good to achieve automated shake-and-catch harvesting, segmenting branches and trunks out as the target object classes. For example, the IoUs for branches and trunks were 0.40 and 0.63, respectively, and the same were 0.78 and 0.96 for apples and leaves, respectively. The results are considered satisfactory because they refer to a 57% and 77% overlap between predicted and ground-truth segments for branches and trunks, meaning that the actual trajectories of branches and trunks can be described within the tolerance of an end-effector of the shaking mechanism. In addition, BFScores of 0.82 and 0.89 were achieved for branches and trunks, also indicating good preservations of their local boundary information.
- When the trained ResNet-18 was tested on images from different crop cultivars and canopy types, it achieved the best results with Pink Lady canopies of light foliage density, as expected, with a mean PcA of 96%, mean IoU of 0.75, and mean BFScore of 0.92 per image basis. Besides, the network performed satisfactorily with images from high foliage density canopies, especially with Scifresh. For example, the IoUs for branches and trunks were 0.41 and 0.62. The BFScores were 0.71 and 0.86 per-class basis in this case, and these are similar to the test results from the original data set of Fuji canopy images discussed. The results showed good robustness of the trained network in automatically identifying the tree branches and trunks across different apple cultivars for automated apple harvesting.
- For modeling the branches and trunks, 3rd-degree polynomial equations were used and achieved R^2 values of 0.40 and 0.67 for branches and trunks, respectively. The polynomial model was then used in detecting shaking points in Fuji canopy images. About 72% of estimated shaking locations were considered "good" with mean errors of 11 pixels along the *y*-axis. The remaining 28% of estimated shaking points had a larger error of approximately 43 pixels compared to manually selected points.

The results generated from this study provide a good foundation for automated shake-and-catch apple harvesting using computer vision systems. With a wide range of canopy scenarios in SNAP tree architectures covered, the data sets from our study can be readily transferred to other similar studies, particularly on medium-/high-density foliage canopies. Furthermore, the computational speed is critical in agricultural robotic applications. With our proposed method, near-real-time processing is possible with a processing speed as fast as 0.35 s per image.

ACKNOWLEDGMENTS

This study was supported partially by USDA Hatch and Multistate Project Funds (Accession Nos. 1005756 and 1001246), a USDA National Institute for Food and Agriculture (NIFA) competitive grant (Accession No. 1005200), and the WSU Agricultural Research Center (ARC). The China Scholarship Council (CSC) sponsored Dr. Xin Zhang in conducting her doctoral dissertation study at the WSU Center for Precision and Automated Agricultural Systems (CPAAS). Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of WSU, USDA, or CSC. We would like to give our thanks to Allan Brothers Fruit Company, Naches, WA, for their great support in field data collection.

ORCID

Xin Zhang http://orcid.org/0000-0001-9654-3859

REFERENCES

Amatya, S., Karkee, M., Gongal, A., Zhang, Q., & Whiting, M. D. (2016). Detection of cherry tree branches with full foliage in planar architecture for automated sweet-cherry harvesting. *Biosystems Engineering*, *146*, 3–15. https://doi.org/10.1016/j.biosystemseng.2015.10.003

Amatya, S., Karkee, M., Zhang, Q., & Whiting, M. D. (2017). Automated detection of branch shaking locations for robotic cherry harvesting using machine vision. *Robotics*, *6*(4), 31. https://doi.org/10.3390/robotics6040031

Bac, C. W., Van Henten, E. J., Hemming, J., & Edan, Y. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, *31*(6), 888–911. https://doi.org/10.1002/rob.21525

Bargoti, S., & Underwood, J. (2017). *Deep fruit detection in orchards. Paper presented at the IEEE International Conference on Robotics and Automation (ICRA '17)*, Marina Bay Sands, Singapore (pp. 3626–3633). https://doi.org/10.1109/ICRA.2017.7989417

Barth, R., IJsselmuiden, J., Hemming, J., & Van Henten, E. J. (2018). Data synthesis methods for semantic segmentation in agriculture: A capsicum annuum dataset. *Computers and Electronics in Agriculture*, *144*, 284–296. https://doi.org/10.1016/j.compag.2017.12.001

Chen, L. C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. https://arxiv.org/abs/1706.05587

Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018, September). Encoder-decoder with atrous separable convolution for semantic image segmentation. *Paper presented at the European Conference on Computer Vision (ECCV '18)*, Munich, Germany (pp. 801–818). https://doi.org/10.1007/978-3-030-01234-2_49

Clark, M. (2017). Washington state's agricultural labor shortage. https://www.washingtonpolicy.org/library/doclib/Clark-Washington-state-s-agricultural-labor-shortage-PB-6-23-17.pdf

Csurka, G., Larlus, D., Perronnin, F., & Meylan, F. (2013). *What is a good evaluation measure for semantic segmentation?* Paper presented at the 24th British Machine Vision Conference (BMVC '13), Bristol, U.K (p. 27). https://doi.org/10.5244/C.27.32

Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Li, F. (2009). *Imagenet: A large-scale hierarchical image database*. Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09), Miami, FL (pp. 248–255). https://doi.org/10.1109/CVPR.2009.5206848

Dias, P. A., Tabb, A., & Medeiros, H. (2018). Multispecies fruit flower detection using a refined semantic segmentation network. *IEEE*

Robotics and Automation Letters, *3*(4), 3003–3010. https://doi.org/10.1109/LRA.2018.2849498

Fu, L., Majeed, Y., Zhang, X., Karkee, M., & Zhang, Q. (2020). Faster R-CNN-based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting. *Biosystems Engineering*, *197*, 245–256. https://doi.org/10.1016/j.biosystemseng.2020.07.007

Gao, F., Fu, L., Zhang, X., Majeed, Y., Li, R., Karkee, M., & Zhang, Q. (2020). Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN. *Computers and Electronics in Agriculture*, *176*, 105634. https://doi.org/10.1016/j.compag.2020.105634

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). *Mask R-CNN*. Paper presented at the IEEE International Conference on Computer Vision (ICCV '17), Venice, Italy (pp. 2961–2969). https://doi.org/10.1109/ICCV.2017.322

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Identity mappings in deep residual networks*. Paper presented at the European Conference on Computer Vision (ECCV '16), Amsterdam, Netherlands (pp. 630–645). https://doi.org/10.1007/978-3-319-46493-0_38

He, L., Zhang, X., Ye, Y., Karkee, M., & Zhang, Q. (2019). Effect of shaking location and duration on mechanical harvesting of fresh market apples. *Applied Engineering in Agriculture*, *35*(2), 175–183. https://doi.org/10.13031/aea.12974

Hohimer, C. J., Wang, H., Bhusal, S., Miller, J., Mo, C., & Karkee, M. (2019). Design and field evaluation of a robotic apple harvesting system with a 3D-printed soft-robotic end-effector. *Transactions of the ASABE*, *62*(2), 405–414. https://doi.org/10.13031/trans.12986

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, *147*, 70–90. https://doi.org/10.1016/j.compag.2018.02.016

Karkee, M., & Adhikari, B. (2015). A method for three-dimensional reconstruction of apple trees for automated pruning. *Transactions of the ASABE*, *58*(3), 565–574. https://doi.org/10.13031/trans.58.10799

Karkee, M., Adhikari, B., Amatya, S., & Zhang, Q. (2014). Identification of pruning branches in tall spindle apple trees for automated pruning. *Computers and Electronics in Agriculture*, *103*, 127–135. https://doi.org/10.1016/j.compag.2014.02.013

Kemker, R., Salvaggio, C., & Kanan, C. (2017). High-resolution multispectral dataset for semantic segmentation. https://arxiv.org/abs/1703.01918

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *Imagenet classification with deep convolutional neural networks*. Paper presented at the Neural Information Processing Systems (NIPS '12), Lake Tahoe, CA (pp. 1097–1105). http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Majeed, Y., Zhang, J.,Zhang, X., Fu, L., Karkee, M., Whiting, M. D., & Zhang, Q. (2020). Deep learning based segmentation for automated training of apple trees on trellis wires. *Computers and Electronics in Agriculture*, *170*, 105277. https://doi.org/10.1016/j.compag.2020.105277

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press.

Peterson, D. L., Bennedsen, B. S., Anger, W. C., & Wolford, S. D. (1999). A systems approach to robotic bulk harvesting of apples. *Transactions of the ASAE*, *42*(4), 871–876. https://doi.org/10.13031/2013.13266

Peterson, D. L., & Wolford, S. D. (2003). Fresh-market quality tree fruit harvester part II: Apples. *Applied Engineering in Agriculture*, *19*(5), 545–548. https://doi.org/10.13031/2013.15314

Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster R-CNN: Towards real-time object detection with region proposal networks*. Paper presented at the Neural Information Processing Systems (NIPS '15), Montréal, Canada (pp. 91–99). http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf

Sa, I., Chen, Z., Popović, M., Khanna, R., Liebisch, F., Nieto, J., & Siegwart, R. (2017). Weednet: Dense semantic weed classification using multispectral images and MAV for smart farming. *IEEE Robotics and Automation Letters*, *3*(1), 588–595. https://doi.org/10.1109/LRA.2017.2774979

Silwal, A., Davidson, J. R., Karkee, M., Mo, C., Zhang, Q., & Lewis, K. (2017). Design, integration, and field evaluation of a robotic apple harvester. *Journal of Field Robotics*, *34*(6), 1140–1159. https://doi.org/10.1002/rob.21715

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. https://arxiv.org/abs/1409.1556

USDA. (2002). S51.300: United States standards for grades of apples. Washington, DC: USDA Agricultural Marketing Service. https://www.ams.usda.gov/grades-standards/applegrades-standards

USDA. (2020). *National agricultural statistics database*. USDA National Agricultural Statistics Service. https://quickstats.nass.usda.gov

Whiting, M. D. (2018). Chapter 6: Precision orchard systems. In Q. Zhang (Ed.), *Automation in Tree Fruit Production: Principles and Practice* (pp. 93–111). CABI.

Zabawa, L., Kicherer, A., Klingbeil, L., Milioto, A., Topfer, R., Kuhlmann, H., & Roscher, R. (2019). *Detection of single grapevine berries in images using fully convolutional neural networks*. Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '19) Workshops, Long Beach, CA. http://openaccess.thecvf.com/content_CVPRW_2019/papers/CVPPP/Zabawa_Detection_of_Single_Grapevine_Berries_in_Images_Using_Fully_Convolutional_CVPRW_2019_paper.pdf

Zhang, J., He, L., Karkee, M., Zhang, Q., Zhang, X., & Gao, Z. (2018). Branch detection for apple trees trained in fruiting wall architecture using depth features and regions-convolutional neural network (R-CNN). *Computers and Electronics in Agriculture*, *155*, 386–393. https://doi.org/10.1016/j.compag.2018.10.029

Zhang, J., Karkee, M., Zhang, Q., Zhang, X., Yaqoob, M., Fu, L., & Wang, S. (2020). Multi-class object detection using faster R-CNN and estimation of shaking locations for automated shake-and-catch apple harvesting. *Computers and Electronics in Agriculture*, *173*, 105384. https://doi.org/10.1016/j.compag.2020.105384

Zhang, X., He, L., Karkee, M., Whiting, M. D., & Zhang, Q. (2020). Field evaluation of targeted shake-and-catch harvesting technologies for fresh market apple. *Transactions of the ASABE*. (In press). https://doi.org/10.13031/trans.13779

How to cite this article: Zhang X, Karkee M, Zhang Q, Whiting MD. Computer vision-based tree trunk and branch identification and shaking points detection in Dense-Foliage canopy for automated harvesting of apples. *J Field Robotics*. 2021;38:476–493. https://doi.org/10.1002/rob.21998