Computers and Electronics in Agriculture 181 (2021) 105933

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Original papers

Canopy segmentation and wire reconstruction for kiwifruit robotic harvesting

Zhenzhen Song [a], Zhongxian Zhou [a], Wenqi Wang [a], Fangfang Gao [a], Longsheng Fu [a,b,c,*], Rui Li [a], Yongjie Cui [a]

[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

ARTICLE INFO

Keywords:
Canopy segmentation
DeepLabV3+
Median frequency weights
Progressive probabilistic Hough transform
Uniform weights
Wire reconstruction

ABSTRACT

Kiwifruits are commercially grown on sturdy support structures such as T-bars and pergolas. Wires are widely used in modern agriculture as an important material for supporting T-bars. It may lead to damage to kiwifruit harvesting end-effector or robot when accessing fruits occluded by branches or wires. Additional development to segment calyxes, branches, and wires will help to achieve higher-level picking strategies. DeepLabV3+ was adopted to segment the fruit calyx, branch, and wire in this work. A method of discrete wire pixels reconstruction was then developed on Progressive probabilistic Hough transform (PPHT) to help sense distribution of the wire. Lines that didn't meet the constraints, i.e., angle or distance between the lines, were regarded as noise and eliminated. There were 327 images divided into training (261) and testing (66) sets, where the training set was augmented to 1566 images. The dataset was heavily imbalanced where the pixels of calyx, branch, and wire were much fewer than background pixels. For the imbalanced kiwifruit canopy image segmentation, it was proven that the uniform weights assignation method outperformed the median frequency weights. In terms of backbone, ResNet-101 achieved IoUs (intersection over union) of 0.686, 0.709, and 0.424 for calyx, branch, and wire, respectively, and the highest mIoU (mean IoU) of 0.694. It took about 210.0 ms to process a resolution of 512 × 341 pixels image, which could be acceptable for the kiwifruit harvesting robot. The PPHT achieved an correct detection rate of 92.4%, and was competitive in processing time of 6.4 ms/image. Canopy image segmentation can provide a basis for guiding the harvesting end-effector to pick kiwifruits safely, thus improving the harvesting success rate and reducing on-orchard costs.

1. Introduction

China is the largest cultivator and producer of kiwifruits in the world, which has been relied on manual fruit picking and thus desiring for mechanical/robotic harvesting. China has a yield of approximately 2.04 million tons kiwifruits in 2018 from a cultivated area of 1.68×10^5 ha (Zhang, 2019a). Within China, the cultivation area of kiwifruits in Shaanxi province was around 5.3×10^4 ha, while the production reached 9.48×10^5 tons, which provides the highest production (UN Food & Agriculture Organization, 2020). Harvesting kiwifruits in this area relies mainly on manual picking, which is a labor-intensive, time-consuming, and high labor cost (Fu et al., 2015). The ageing population and increasing labor cost have led to decrease in available labor force for agriculture harvesting (Zhou et al., 2016). Autonomous mechanized solutions to supplement manual labor is an emerging strategic necessity. Therefore, there is a strong desire to introduce mechanical/robotic harvesting.

Kiwifruit orchards use a pergola growing system with wires that support fruit and canes in an overhead roof-type structure as T-bar, which is conducive to mechanical harvesting operations than other tree fruits in China. The low-cost T-bar has been widely used in China (Fu et al., 2019, 2020a; Zhou et al., 2020). Posts, with near uniform spacing support the overhead structure, dividing the canopy area into rows. It consists of a 1.7 m high post and an approximately 1.7 m wide cross arm, variations in infrastructure are often found in width according to the shape and size of the orchard. The wires run on top of the cross arms and are connected in the middle and on both edges of the cross arms. They are formed in a grid with a spacing of 50 cm. The upper stems of the

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

https://doi.org/10.1016/j.compag.2020.105933
Received 29 August 2020; Received in revised form 23 November 2020; Accepted 28 November 2020
Available online 25 December 2020
0168-1699/© 2020 Elsevier B.V. All rights reserved.

kiwifruit are tied to the top wires so that the egg-sized kiwifruits would be hanging downward, which makes them visible and accessible for picking (Liu et al., 2020; Mu et al., 2020). This canopy structure also provides a relatively simpler and structured workspace for mechanized or automated field operations such as robotic harvesting compared to other fruit trees in China, such as apples.

Accurately and robustly segment semantic information from the working scene in orchard is necessary for fruit robotic harvesting. Remotely positioning the camera underneath canopy allows fruit segmentation while harvesting over a larger canopy area. There are four classes objects in canopy images, i.e., fruits, leaves, branches, and wires, that a kiwifruit harvesting end-effector may connect during on-orchard working. Leaves are soft to be accessed safely. While branches and wires are rigid, it may lead to damage to the end-effector or robot when the end-effector is entangled or collided with them (Fu et al., 2020b). Fruits are picking targets and maybe adjacent or overlapped on canopy images, where the calyx of each kiwifruit is visible and independent. There is a reflective bright area and distinguishable border around the fruit calyx, making the calyx more distinctive (contrast increase). It seems more feasible to utilize a semantic segmentation algorithm to separate the calyx areas than to segment the fruit. Besides, it is desirable to position the end-effector to the center of the fruit calyx (Fu et al., 2019). Therefore, the fruit calyx can be detected on canopy images for robotic harvesting. Additional development to segment calyxes, branches, and supporting wires will help to achieve higher-level picking strategies. However, color of the calyx and branch is similar to the background, such as hay and leaves. Besides, wires in orchard are too thin to be easily segmented from a far view image. Therefore, it is crucial to develop a highly effective and robust vision algorithm to accurately segment calyx, branch, and wire under the complex on-orchard environments.

Deep learning as a powerful technique in the artificial intelligence field is becoming a prevalent way of semantic segmentation in agricultural fields, and applied to segment images acquired in orchard. Santos et al. (2020) adopted Mask R-CNN (Region-based Convolutional Neural Network) with ResNet-101 (Residual Network with 101-layer) to segment grape clusters and reached an F1-score of 0.847 for instance segmentation on intersection over union (IoU) of 0.5. Lin et al. (2019) applied FCN (Fully Convolutional Neural Networks) to perform the segmentation of guava fruits and branches and realized IoUs of 0.806 and 0.473 for fruit and branch, respectively, and cost 0.165 s for branch segmentation on a 424 × 512 pixels image with a GPU of NVIDIA GTX-1060. Yu et al. (2019) used Mask R-CNN to detect strawberry, which achieved a mean IoU (mIoU) of 0.899 for ripe and unripe fruit and spent 0.125 s in average to process a 640 × 480 pixels image with a GPU of NVIDIA GTX-1080. Liang et al. (2020) utilized a U-Net (U-shape network designed based on FCN) to segment the branches of litchi and realized an IoU of 0.739 under nighttime natural environment. Majeed et al. (2020b) used SegNet (Semantic Segmentation Network) with VGG16 (Visual Geometry Group with 16 layers) for median frequency weights to segment trunk, branch, and wire in apple orchards and obtained IoUs of 0.580, 0.350, and 0.260 for trunk, branches, and wire, respectively. For kiwifruit orchards, only Williams et al. (2019) tried FCN to segment calyx, branch, and wire and cost 3 s on a 1920 × 1200 pixels image with a GPU of NVIDIA GTX-1080, and there were some branches and wires missed or discrete segmentation on canopy images. It was generally reported that FCN is demonstrated to be feasible on kiwifruit canopy images but slow speed, and thus being replaced by other object detection networks (Williams et al., 2020). However, Mask R-CNN, U-Net, and SegNet have been proven to achieve higher IoUs in recent years. Therefore, it is promising to use a new semantic segmentation network to segment calyx, branch, and wire in kiwifruit orchards.

DeepLabV3+ proposed as a new semantic segmentation network has been proven to perform better than conventional deep learning methods such as SegNet, U-Net, and FCN. Zhang et al. (2020a) developed a semantic segmentation algorithm with DeepLabV3+ and super-pixel segmentation algorithm-quick shift and achieved mIoU of 0.692 on PASCAL VOC2012, which revealed that the proposed method could provide a more efficient solution. Ayhan and Kwan (2020) utilized DeepLabV3+ with Xception to classify forest, grassland, and shrub land, which obtained mIoU of 0.607 for uniform weights and 0.582 for median frequency weights on Slovenia dataset, and cost 360 ms on a 499 × 505 pixels image using an RTX2070 GPU. Kang and Chen (2020) applied DeepLabV3+ with ResNet-101 to perform segmentation on fruits and branches in apple orchard, which achieved IoU of 0.873 for apples, and 0.794 for branches, and an average computational time of 70 ms to process a 640 × 480 pixels image on an NVIDIA GTX-1080Ti GPU. Zhang et al. (2019b) used DeepLabV3+ with ResNet-18 to segment apple, leaves, and branch using median frequency weights, which achieved IoUs of 0.408, 0.717, and 0.944 for branch, apples, and leaves, respectively, and cost 0.33 s on a 512 × 424 pixels image of apple trees with a GPU of NVIDIA GTX-1080. Those studies showed that DeepLabV3+ is promising in speed and accuracy for semantic segmentation.

Besides, wires discrete appeared on canopy images need to be reconstructed to obtain all the information of wires. Some branches and wires are behind fruits or leaves, and thus discrete appearing on canopy images (Fu et al., 2020c; Gao et al., 2020; Ji et al., 2017; Zhang et al., 2020b). Moreover, proportion of wire pixels in an image (0.6%) was much smaller than that of branches (9.0%) in this work, as mentioned in section 3.1, which may make the segmentation of wires more difficult than branches. Therefore, the discrete pixels of wires on canopy images need to be reconstructed. Progressive probabilistic Hough transform (PPHT) is a probabilistic algorithm on line extraction and widely used in the field of lane line detection field. Marzougui et al. (2020) employed PPHT to detect lane markings, which achieved a correct detection rate (CDR) of 93.8% and cost 21.5 ms on a 640 × 480 pixels image with an Intel Core i7-2630QM CPU. Chang and Kang (2018) applied the PPHT to detect lane markings by removing lanes that didn't reach requirements of set angle or distance conditions. Chen et al. (2020) utilized PPHT to detect welding flame and achieved CDR of 90.2%. Straight lines in the field of lane line detection researches are presented as regular or dashed, which are easy to be detected because the orientation angles of the left and right lane lines in the dataset images remain basically the same. However, the angles of the wire lines seem to be messy in kiwifruit canopy images. Besides, there was no report of linear object detection in complex orchard. Therefore, further research on discrete pixels of wires could be performed based on the lane line detection.

DeepLabV3+ was employed to segment calyx, branch, and wire in pixel-level on canopy images acquired in kiwifruit orchards to sense the distribution of calyx, branch, and wire and guide the harvesting end-effector to approach fruits safely. The remainder of this paper was structured as follows: Section 2 presented the image acquisition, image dataset and annotation, data augmentation, and the methodology, including a description of DeepLabV3+ with different backbones, two methods of weights assignation including uniform weights and median frequency weights during network training used for canopy image segmentation, a method of reconstructing the discrete pixels for wires, and evaluation indicators. Section 3 discussed the results; Finally, Section 4 presented the conclusions obtained in this work.

2. Materials and methods

2.1. Image acquisition

Kiwifruit images were captured from Meixian Kiwifruit Experimental Station (latitude: 34°07′39″N, longitude: 107°59′50″E, and 648 m in altitude, cv. 'Hayward') in Shaanxi province, China. The kiwifruit trees were planted in 2006. The method of capturing images used in this work was based on placing a camera underneath canopy, with its central axis perpendicular to the canopy, as shown in Fig. 1. A common single-lens reflex camera (Canon S110, Canon Inc., Tokyo, Japan) with a 1/1.7 in. sensor on 'AUTO' mode with a resolution of 2352 × 1568 pixels was mounted on a mobile tripod at around 100 cm below the fruit surface to

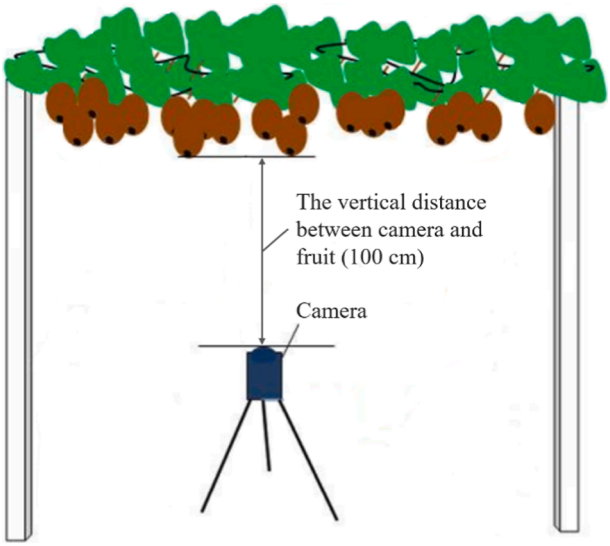


Fig. 1. Schematic diagram of image capturing system.

optimize the number of kiwifruits included in the field of view of the camera (94.2°), where the canopy area covered by one single shot of the camera was around 70 cm × 105 cm. RGB (Red, Green, and Blue) images were captured from three harvesting seasons of 2016, 2017, and 2018. The images were taken at different times of the day without artificial lighting and saved in JPEG format.

2.2. Image dataset and annotation

A total of 327 images were used to form original dataset. All the original RGB images were scaled to 512 × 341 pixels by a linear interpolation algorithm in advance to reduce the running time of subsequent experiments and the limitation of GPU memory. LabelMe 4.5.6 (an image labeler toolbox developed by MIT's Computer Science and Artificial Intelligence Laboratory) was used to label the canopy images. And each image was hand-labelled in pixel-level with four classes: calyx, branch, wire, and background. The corresponding annotated files were saved as JSON format and then parsed into PNG format images which were called ground truth images. An example of the RGB image and its corresponding ground truth image were shown in Fig. 2a and Fig. 2b, respectively. The 'polygon' option was applied to label the outline of calyx, branch or wire and give each individual class a separate label. Some petioles were labelled as background because they would not cause damage to the end-effector or robot (Gao et al., 2020). Only the visible areas of calyx, branch, and wire were labeled with their corresponding classes. The labelled dataset was divided into training (261) and testing (66) sets at a ratio close to 4:1. The training set was randomly obtained from the independent and uniform sampling of the whole dataset. The training set and testing set were mutually exclusive, which ensured the reliability of the later evaluation standards. The number of the training set was not enough to train DeepLabV3+ model, and data augmentation techniques were thus applied in Section 2.3.

2.3. Data augmentation

To improve the overall learning procedure and performance, data augmentation techniques were used to artificially enlarge the number of training set, by means of feeding a model with varied data. In this section, data augmentation methods including image rotation in 90°, 180°, and 270° and image mirroring in horizontal and vertical axis on the training set, were implemented using the software MATLAB (ver. 2018b, The MathWorks, Natick, MA, USA) with the Image Processing Toolbox in this work. For the image rotation, MATLAB function '*imrotate*' was used to rotate the training set with 90°, 180°, and 270° by changing its parameter '*angle*', respectively. Image mirroring (in horizontal and vertical axis) was carried out using the function '*flip*' in MATLAB (Luus et al., 2015). The horizontal mirror images and the vertical mirror images were obtained by setting the flip parameter '*dim*' to '1' and '2', respectively. The training set (261) including RGB image along with its corresponding ground truth image was augmented to 1566 images by the five above mentioned methods.

2.4. DeepLabV3+ with different backbones

DeepLabV3+ is an encoder-decoder module with atrous separable convolution for semantic image segmentation and provided very promising results in the PASCAL VOC2012 data challenge (Chen et al., 2018). It contains multi-scale semantic information from the encoder module, while the object boundaries are recovered by the decoder module. The block diagram of DeepLabV3+ was shown in Fig. 3. It uses Atrous Spatial Pyramid Pooling (ASPP) mechanism, which probes convolutional features at multiple scales by applying atrous convolution with different rates of image-level features. At the encoder module, the training set is fed into a backbone to obtain a feature map, and the feature map is then inputted into the ASPP, which consists of a 1 × 1 convolution, three 3 × 3 atrous convolutions (with atrous rates as 6, 12, and 18), and an image pooling layer. All the feature maps of the ASPP were concatenated to generate the combined feature map. Then, a 1 × 1 convolution layer is applied to fuse the information within the combined feature map and reduce the channel number of the feature maps. The last feature map after 1 × 1 convolution is used as the encoder output

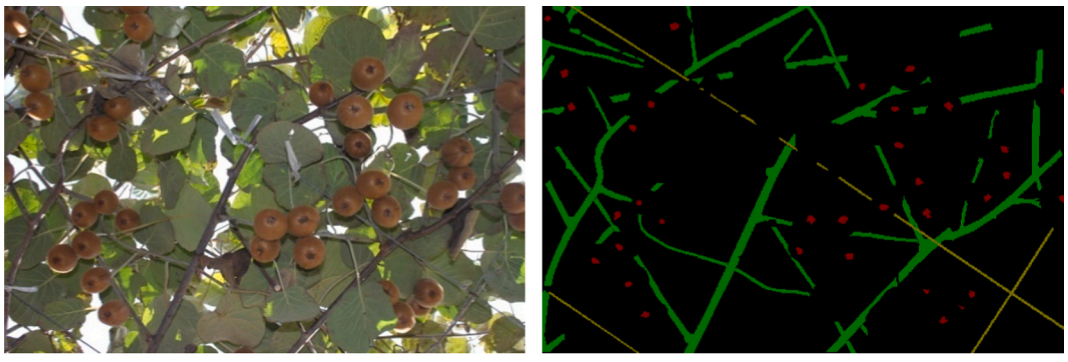


Fig. 2. Examples of an RGB image in kiwifruit orchards (a) and its ground truth with 'polygon' (b). Black represented the pixel of background, red represented the pixel of calyx, green showed the pixel of branch, and yellow represented wire. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

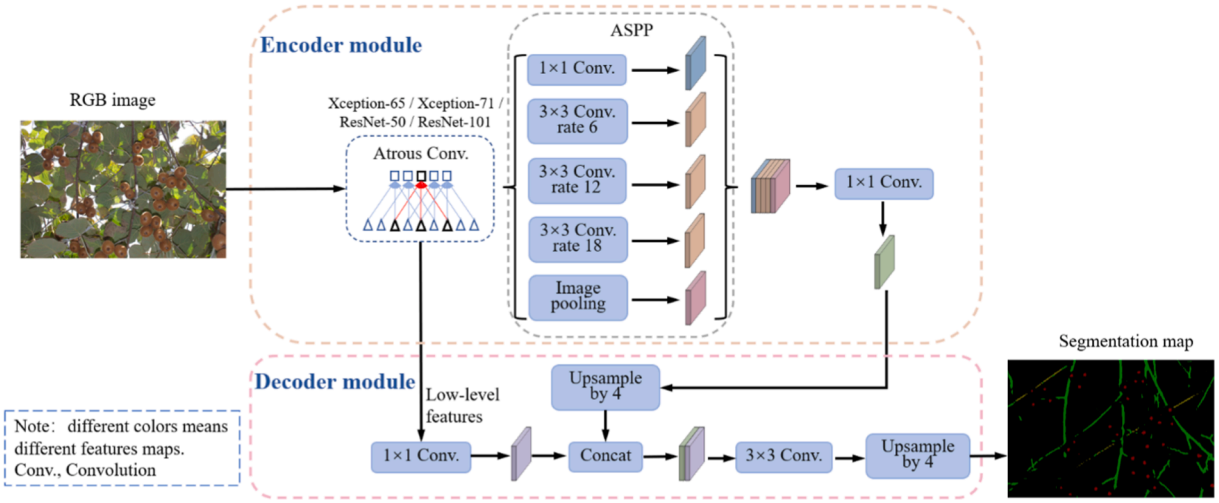


Fig. 3. Block diagram of DeepLabV3+ for kiwifruit canopy image segmentation.

module. The encoder features are first bilinearly upsampled for four times and then concatenated with the corresponding low-level features from the backbone that has the same spatial resolution. Another 1 × 1 convolution on the low-level features is applied to reduce the number of channels. After the concatenation, a 3 × 3 convolution is used to refine the features followed by another simple bilinear upsampling for four times.

To develop a better segmentation model for kiwifruit orchard image, four backbones, i.e., Xception-65, Xception-71, ResNet-50, and ResNet-101, were trained based on DeepLabV3+ in this work, as shown in Fig. 3. Xception, a convolutional neural network architecture (Chollet, 2017) based entirely on depthwise separable convolution layers, is adapted in three respects to the task of semantic segmentation of DeepLabV3+ . The three respects are replacing all max-pooling layers by atrous separable convolutions, adding extra batch-norm and ReLU after each 3 × 3 depthwise convolution, and more layers. ResNet, a residual neural network (He et al., 2016) proposed to combat the vanishing gradient problem during training deep convolutional networks, is modified for using in DeepLabV3+ by replacing the first 7 × 7 convolution with three 3 × 3 convolutions. The modified Xception and ResNet are improved at different numbers of convolutional layers, i.e., Xception-65, Xception-71, ResNet-50, and ResNet-101.

2.5. Network training and hyperparameters

Training architecture for kiwifruit canopy image segmentation based on DeepLabV3+ was illustrated in Fig. 3. Transfer learning can quickly adapt to new tasks in the event of insufficient dataset. Using pre-trained model with a large dataset, the underlying structure weight parameters can be shared, and the model can be fine-tuned to overcome the differences between the pre-training and new datasets. In this work, weights of a pre-trained model except for the final classification layer were used for initialization and fine-tuning. The hyperparameters of DeepLabV3+ used in this work were selected after fourteen trials, as seen in Table 1. A stochastic gradient descent algorithm was used to find the minimum loss in the network during training, which was expected to help converge the training process easily (Keskar et al., 2017). The models were trained end-to-end without piecewise pre-training of each component.

In this work, DeepLabV3+ model was trained with two weights assignation methods i.e. uniform weights and median frequency weights, in the cross-entropy loss. The uniform weights are set the weights of the four classes, i.e., background, calyx, branch, and wire, to the same value of 1.0. The median frequency weights are set according to the proportion of pixels of the four classes in an image (Dong et al., 2019; Majeed et al., 2020a). The median frequency weight of class i (abbreviated as w_i) referred to the ratio of the frequency of pixels in class i to the median value of each f_i, as in Eq. (1). The f_i represented the number of pixels of the class i divided by the total number of pixels in images where i was present. The *median* (f_i) was the median of these frequencies. The value i represented each class on canopy images in this work: background ($i = 1$), calyx ($i = 2$), branch ($i = 3$), and wire ($i = 4$).

$$w_i = \frac{median(f_i)}{f_i} \tag{1}$$

Table 1
Training parameters used in DeepLabV3 + .

Training parameters	Values
Model variant	Xception-65 / Xception-71 / ResNet-50 / ResNet-101
Base learning rate	0.005
Weight decay	0.00004
Training number of steps	50,000
Learning policy	Poly
Train crop size	[513, 513]
Learning rate decay step	2000
Decoder output stride	4
Atrous rates	[6, 12, 18]
Train output stride	16
Momentum	0.9
Train batch size	4
Fine tune batch norm	False
Initialize last layer	False
Tf initial checkpoint	deeplabv3_pascal_train_aug

The training and testing of DeepLabV3+ model platform included a desktop computer (shgentai SP16HDIET, www.Shgentai.com) with Intel Xeon E5-1650 v4 (3.60 GHz) six-core CPU, and a GPU of NVIDIA TITAN XP 12 GB GPU (3840 CUDA cores) and 32 GB of memory, running on an Ubuntu 16.04 64 bit system. The software tools included CUDA 9.0, CUDNN 7.1, and Python 3.6. The experiments were implemented in the TensorFlow 1.10.0 framework.

2.6. Reconstruction of the wire

After segmentation, the pixels of wire were discrete appeared on the segmentation maps, which couldn't help sense the distribution of the wire in orchards. Therefore, it was necessary to connect the discrete points or short line areas to reconstruct the wires on the canopy images. Similar to lane line detection in Marzougui et al. (2020), further work based on the PPHT algorithm (Viola and Jones, 2001) was performed to reconstruct the discrete areas of wires.

PPHT is a probabilistic algorithm on line extraction using a random sampling of the edge points. It is used to enhance memory efficiency and improve computational time for its' accumulating only a small fraction of pixels as candidates of the wire lines. The straight lines can be recognized more accurately with PPHT than standard Hough transformation (Chen et al., 2020; Feng et al., 2020). The reconstruction of wires was carried out on an Intel Xeon E5-1650 v4 (3.60 GHz) six-core CPU in this work, using the OpenCV and Python language. The flowchart of PPHT was shown in Fig. 4. Some pre-processing operations were applied before using the '*cv2.HoughLinesP*' function in the OpenCV library. First, the discrete and discontinuous pixels of wires on the segmentation maps were extracted. Then binary and skeleton refinement operations were implemented to achieve faster processing and a smaller memory footprint. A skeleton is implemented by the *skimage.morphology* module in Python preserved the structure of the shape, but all redundant pixels should be removed. After that, the skeletonized images were fed into the '*cv2.HoughLinesP*' as input. The argument of *threshold* that represented the minimum number of intersections to 'detect' a line was set as 20. The minimum number of points that form a line (*minLinLength*) was assigned as 30, namely, lines with less than 30 points would be disregarded. The maximum gap between two points (*maxLineGap*) to be considered in the same line was 200 to connect the short line segment. According to the two coordinates (x_{start}, y_{start}, x_{end}, y_{end}) returned by the PPHT, the detected lines were extended to the entire image.

PPHT does not consider the pre-defined number of voting pixels in the accumulator. This leads to unwanted lines for various reasons, such as the insufficient amount of lines and occlusion by other objects on the segmentation maps. Wires are straight and approximately parallel or perpendicular to the other on the canopy images. The angle threshold was set $\pm$ 20° between two approximately parallel lines or the perpendicular of one line and the other. Besides, the distance between two parallel lines exceeds 50 pixels in this work. Therefore, the angle or distance among the detected lines were used as constraints and added to the PPHT. Lines that didn't meet the constraints mentioned above were regarded as noise and eliminated.

2.7. Performance evaluation

Performance of segmentation was evaluated using Pixel Accuracy (PA), IoU, and mIoU in the testing set. PA simply found the ratio of correctly classified pixels divided by the total number of pixels, as shown in Eq. (2). Used as one of the most common metrics in semantic segmentation, IoU was defined as the area of intersection between the predicted segmentation map and the ground truth, divided by the area of union between them, as shown in Eq. (3). The mIoU was another popular metric, which was defined as the average IoU over all classes. And speed was used to evaluate the segmentation time of the model.

$$PA = \frac{\sum_{i=0}^{K} p_{ii}}{\sum_{i=0}^{K}\sum_{i=0}^{K} p_{ij}} \tag{2}$$

$$IoU = J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

where p_{ij} was the number of pixels of class i in ground truth images predicted as class j on segmentation maps; A and B denoted the ground truth images and the predicted segmentation maps, respectively. The value of IoU ranged between 0.0 and 1.0.

Similar to lane line detection (Marzougui et al., 2020), three criteria were used to evaluate the performance of wire line detection, i.e., missed detection rate (MDR), incorrect detection rate (IDR), and correct detection rate (CDR). MDR referred to the percentage of the images where a line was missed detected in the testing set. IDR referred to the percentage of the images where an incorrect detected line was present in the testing set. CDR referred to the percentage of the images where lines were correctly detected in the testing set. The MDR, IDR, and CDR were calculated using Eq. (4), Eq. (5), and Eq. (6), respectively.

$$MDR = \frac{MD}{N_{ALL}} \times 100\% \tag{4}$$

$$IDR = \frac{ID}{N_{ALL}} \times 100\% \tag{5}$$

$$CDR = \frac{CD}{N_{ALL}} \times 100\% \tag{6}$$

where missed detection (*MD*) was the number of images where a line was missed, incorrect detection (*ID*) was the number of images where a line was incorrectly detected, correct detection (*CD*) was the number of images where lines were correctly detected in the testing set, and N_{All} was the number of images in the testing set.

3. Results and discussion

3.1. Weights assignation

The number of pixels for each class, which was denoted by 'pixel count', its frequency (f_i), median frequency weights (w_i), and uniform weights for the training set can be seen in Table 2. The uniform weights of the four classes, i.e., background, calyx, branch, and wire, were set to the same value of 1.0. The w_i of each class i was computed by Eq. (1). From Table 2, the number of pixels for some of the classes, such as calyx, branch, and wire were quite low. The pixels of the background were about 81, 10, and 164 times more than that of calyx, branch, and wire. For avoiding such a bias, the median frequency weights of background, calyx, branch, and wire were assigned with 0.06, 4.60, 0.56, and 9.28 during the fine-tuning of the model, respectively. After median frequency weighting, the calyx, branch, and wire had heavier weighting.

Table 2
Pixel numbers for each class, the median frequency weights and uniform weights used in DeepLabV3+ .

	background	calyx	branch	wire
Pixel count	244,265,451	3,004,787	24,648,008	1,490,090
Frequency (f_i)	0.893	0.011	0.090	0.006
Median frequency weights (w_i)	0.06	4.60	0.56	9.28
Uniform weights	1.0	1.0	1.0	1.0

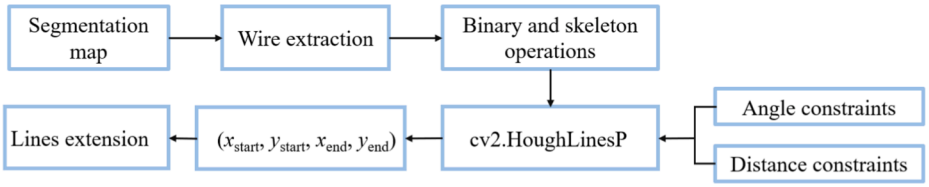


Fig. 4. Flowchart of PPHT for the discrete wire pixels on kiwifruit segmentation maps.

3.2. Performances of DeepLabV3+ models

3.2.1. Training assessment

The training loss curves of the eight DeepLabV3+ models based on Xception-65, Xception-71, ResNet-50, and ResNet-101 with two weights assignation methods were shown in Fig. 5. Different colors and line types represented different backbones and different weight assignation methods in Fig. 5, respectively. The training number of iterations for the four backbones was set as 50,000. The loss values of the eight models decreased as the number of steps increased, and converged at low values when the number of steps reached about 20,000 for Xception-65 / Xception-71, and 40,000 steps for ResNet-50 / ResNet-101, respectively. It was an indicator of a sufficiently trained process both in median frequency weights and uniform weights. And the lowest value of Xception-65, Xception-71, ResNet-50, and ResNet-101 assigned with uniform weights reached 0.192, 0.423, 0.486, and 0.504, respectively. The lowest value in each of the four backbones (in the same sequence) trained with median frequency weights reached 0.144, 0.366, 0.440, and 0.450, respectively. As expected, the initial training losses of the models trained with median frequency weights were lower than those assigned with uniform weights. The low loss values demonstrated that the eight models adopted in this work efficiently learned the features with good convergence ability, which were promising to achieve desired results.

3.2.2. Performance of different weights assignation methods

DeepLabV3+ models assigned with uniform weights outperformed than those assigned with median frequency weights in this work. Taking ResNet-101 as an example, DeepLabV3+ models with uniform weights and median frequency weights obtained the mIoUs of 0.694 and 0.480, respectively, as shown in Table 3. With the use of uniform weights when training DeepLabV3+ model with ResNet-101, the PA increased from 87.6% to 96.0%, similar to other backbones. Even though using median frequency weights helped to reduce the loss value of the models, the PA and mIoU decreased in median frequency weights in this work.

DeepLabV3+ model assigned with uniform weights showed a better segmentation performance on canopy images, clearer in the border area, and fewer misclassified pixels, as shown in Fig. 6b. Fig. 6 presented an example of the segmentation maps of ResNet-101 with two weights assignation methods in the testing set. It showed that a canopy image was correctly segmented into four classes: 'background' (black areas), 'calyx' (red areas), 'branch' (green areas), and 'wire' (yellow areas). With respect to the calyx, branch, and wire, the segmented areas seemed bigger than those in the ground truth image (Fig. 6a) when the model

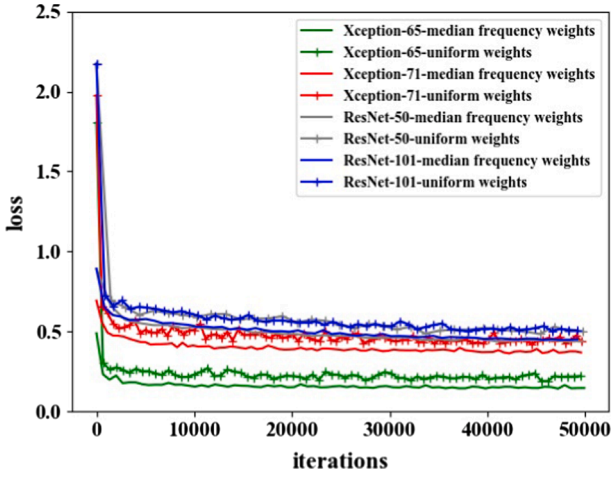


Fig. 5. Training loss curves of DeepLabV3+ models based on Xception-65, Xception-71, ResNet-50, and ResNet-101 with uniform weights and median frequency weights, respectively.

Table 3
Performance of DeepLabV3+ assigned with uniform weights and median frequency weights.

Weights assignation methods	Backbone	mIoU	PA
Uniform weights	Xception-65	0.678	96.1%
	Xception-71	0.640	95.7%
	ResNet-50	0.676	96.0%
	ResNet-101	**0.694**	96.0%
Median frequency weights	Xception-65	0.426	88.4%
	Xception-71	0.465	86.4%
	ResNet-50	0.482	87.6%
	ResNet-101	0.480	87.6%

was assigned with median frequency weights, as shown in Fig. 6c. And when Fig. 6b and Fig. 6c being compared with Fig. 6a, it was noticed that the background pixels misclassified as calyx, branch, or wire in Fig. 6c were more than those in Fig. 6b. The segmentation error could mainly be attributed to difficultly in distinguishing the color of calyx, branch, wire, and background even with human eyes. This led to the deviation of calyx, branch, and wire labels in the training set (Yu et al., 2019). Therefore, it was proven that the uniform weights assignation method outperformed the median frequency weights for the calyx, branch, and wire segmentation on the kiwifruit canopy images during network training.

3.2.3. Performance of different backbones

In terms of backbone, ResNet-101 achieved the highest mIoU for segmentation. The segmentation performance of DeepLabV3+ models with the four backbones assigned with uniform weights was presented in Table 4. ResNet-101 obtained the highest performance with mIoU of 0.694, outperformed than Xception-65 (0.678), ResNet-50 (0.676), and Xception-71 (0.640). And by replacing ResNet-50 with a deeper network ResNet-101, ResNet-101 got an additional 1.8% improvement in mIoU. There was a similar improvement in mIoU for branch segmentation (0.6%) in apple orchards (Kang and Chen, 2020), which obtained IoUs of 0.788 and 0.794 for ResNet-50 and ResNet-101, respectively. With respect to different backbones, ResNet-50, ResNet-101, Xception-65, and Xception-71 achieved similar PA of 96.0%, 96.0%, 96.1%, and 95.7% in the same testing set.

ResNet-101 achieved the highest IoUs of 0.709 and 0.424 for the branch and wire, respectively. In terms of calyx segmentation, ResNet-101 and Xception-65 both achieved the similar highest IoUs of 0.686 and 0.686, respectively. It was significant to ensure that the pixels of the branch and wire were segmented as much as possible for sensing the distribution of branch and wire, and thus to achieve higher-level picking strategies for the harvesting end-effector could avoid entangling or colliding with them. Fig. 7 presented an example of segmentation maps and overlaps of DeepLabV3+ based on four backbones with uniform weights, where ResNet-101 showed the best segmentation result.

ResNet-101 took about 210.0 ms on average to segment calyx, branch, and wire in one image with the resolution of 512 × 341 pixels. The harvesting system was intended to harvest 4 fruit/s (Williams et al. 2019). Therefore, it was acceptable for kiwifruit robotic harvesting system in the variable and complex on-orchard environments to cost 210.0 ms to segment a canopy image. From Table 4, ResNet-50 achieved the fastest segmentation speed of 170.0 ms, followed by ResNet-101, Xception-65, and Xception-71, which cost 210.0, 259.8, and 264.2 ms to process an image, respectively. Analysis of variance (ANOVA) was used to compare the speed of the DeepLabV3+ model for different backbones at a significance level of 0.05 based on Tukey's Honest significant difference test. From Table 4, there was a significant difference in segmentation speed among Xception-71 (or Xception-65), ResNet-50, and ResNet-101, while no significant difference between Xception-71 and Xception-65. The reason was that ResNet-101 had more convolutional layers and parameters than ResNet-50 that resulted in a large trained weights size of 349 MB, which was more than 1.7 times as of

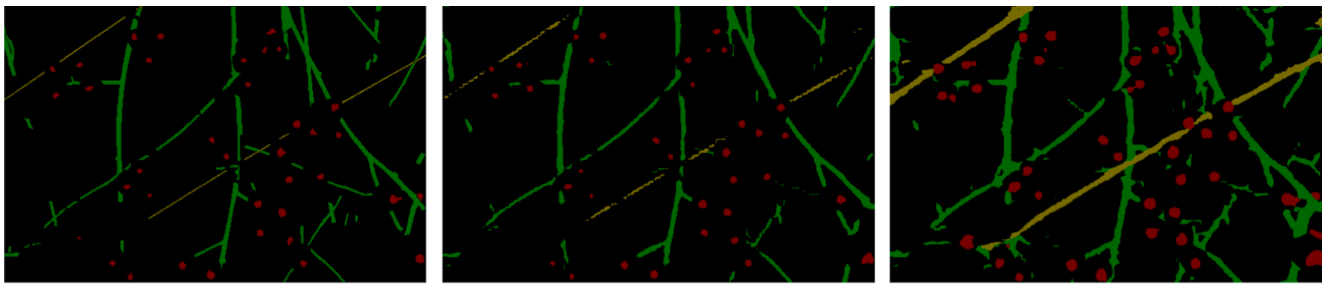


Fig. 6. Comparison of the ground truth image (a), the segmentation maps of DeepLabV3+ based on ResNet-101 with the uniform weight (b) and median frequency weights (c). The black, red, green, and yellow pixels represented the areas of background, calyx, branch, and wire, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
Performance of DeepLabV3+ model assigned with uniform weights with different backbones.

Backbone	IoU				mIoU	PA	Speed	Model size (MB)
	Background	Calyx	Branch	Wire			(ms / image)	
Xception-71	0.954	0.678	0.680	0.248	0.640	95.7%	264.2 ± 3.9[a]	332
Xception-65	0.957	**0.686**	0.709	0.362	0.678	96.1%	259.8 ± 3.5[a]	313
ResNet-101	0.956	0.686	**0.709**	**0.424**	**0.694**	96.0%	210.0 ± 3.4[b]	349
ResNet-50	0.957	0.683	0.701	0.361	0.676	96.0%	170.0 ± 4.3[c]	204

Note: Same letters in the 'speed' column represent no significant difference at the 0.05 level.

ResNet-50 (204 MB), similar to Xception-65 (313 MB) and Xception-71 (332 MB). As can be seen from Table 4, the deeper the backbone was, the more complex the model structure was, and the slower the segmentation speed was. As long as enough labeled data employed for training, the increased model size and computational cost tend to translate to immediate quality gains for most tasks (Szegedy et al. 2016). Therefore, DeepLabV3+ model based on ResNet-101 can be used for the segmentation of calyx, branch, and wire in kiwifruit orchards.

3.2.4. Comparison with other methods

Comparing with the previous semantic segmentation methods for kiwifruit orchards (Williams et al. 2019), DeepLabV3+ model based on ResNet-101 assigned with uniform weights showed the best segmentation performance for calyx, branch, and wire on canopy images. There were some wires missed or discrete segmented on kiwifruit canopy image in Williams et al. (2019), where its low segmentation result cannot help to sense the distribution wire on canopy images completely. And if the harvesting robot ignored those wires, which the end-effector would connect, it would lead to damage to the harvesting end-effector and failure of the picking process (Ji et al., 2017). Moreover, Williams et al. (2019) spent 3 s on a 1920 × 1200 pixels image with an NVIDIA GTX-1080 GPU, which preventing the system from operating at 4 fruit/s (Williams et al., 2020). A conclusion can be drawn that DeepLabV3+ model based on ResNet-101 was competitive in speed, which took about only 210.0 ms on an image with a resolution of 512 × 341 with a GPU of NVIDIA TITAN XP 12 GB GPU.

3.3. Performance of wire reconstruction

The PPHT method applied in this work showed good performance on the reconstruction of segmented wire pixels. The 66 images in the testing set were used to evaluate the detection performance of PPHT. Fig. 8 presented an example of wire segmentation map from original kiwifruit canopy image and its reconstruction image. After processing by the PPHT, the number of images where lines of wire correctly detected (*CD*) was 61, the number of images where a line was missed detected (*MD*) was 4, and the number of images where an incorrect detected line was present (*ID*) was 1. As shown in Table 5, The *CDR* achieved by the PPHT was 92.4%, *MDR* was 6.1%, and *IDR* was 1.5%. The line detection process of a segmentation map with a resolution of 512 × 341 took about 6.40 ms and outperformed the lane markings detection of 21.5 ms on a 640 × 480 pixels image operated on an Intel Core i7-2630QM CPU with 8 GB RAM (Marzougui et al., 2020). Therefore, the PPHT method of reconstructing the discrete wires can be used to help the kiwifruit robotic harvesting systems sense the distribution of wires, including the class of each pixel and its location in kiwifruit canopy image, which can provide a basis for guiding the harvesting end-effector to pick kiwifruits safely.

4. Conclusions

The segmentation of kiwifruit calyx, branch, and wire was very important, as they could be used to sense the distribution of calyx, branch, and wire and guide the harvesting end-effector to approach fruits safely. A new and faster semantic segmentation network DeepLabV3+ was applied in this work, which was proven to achieve a better segmentation on calyx, branch, and wire in pixel-level in this work. And a novel methodology for the calyx, branch, and wire segmentation in kiwifruit images using uniform weights during network training was presented.

In terms of backbone, ResNet-101 achieved mIoU of 0.694, and PA of 96.0% for kiwifruit canopy images, which outperformed other backbones, i.e., ResNet-50, Xception-65 and Xception-71. The IoUs for calyx, branch, and wire in ResNet-101 were 0.686, 0.709, 0.424, respectively. It cost about 210.0 ms to process a resolution of 512 × 341 pixels image, which could be acceptable for the kiwifruit harvesting robot to segment the canopy images acquired in the complex on-orchard environments.

To help sense the wires distribution on the segmentation maps, a line-detection method based on PPHT was utilized to reconstruct the discrete wires on the segmentation maps. The method achieved a CDR of 92.4%, MDR of 6.1%, and IDR of 1.5% for the line detection of wire. It only took about 6.4 ms to process a 512 × 341 image on Intel Xeon E5-1650 v4 (3.60 GHz) six-core CPU, which could realize a real-time application.

Finally, the way how to compress model size, for example, a quantization method, improve segmentation speed, robustness while

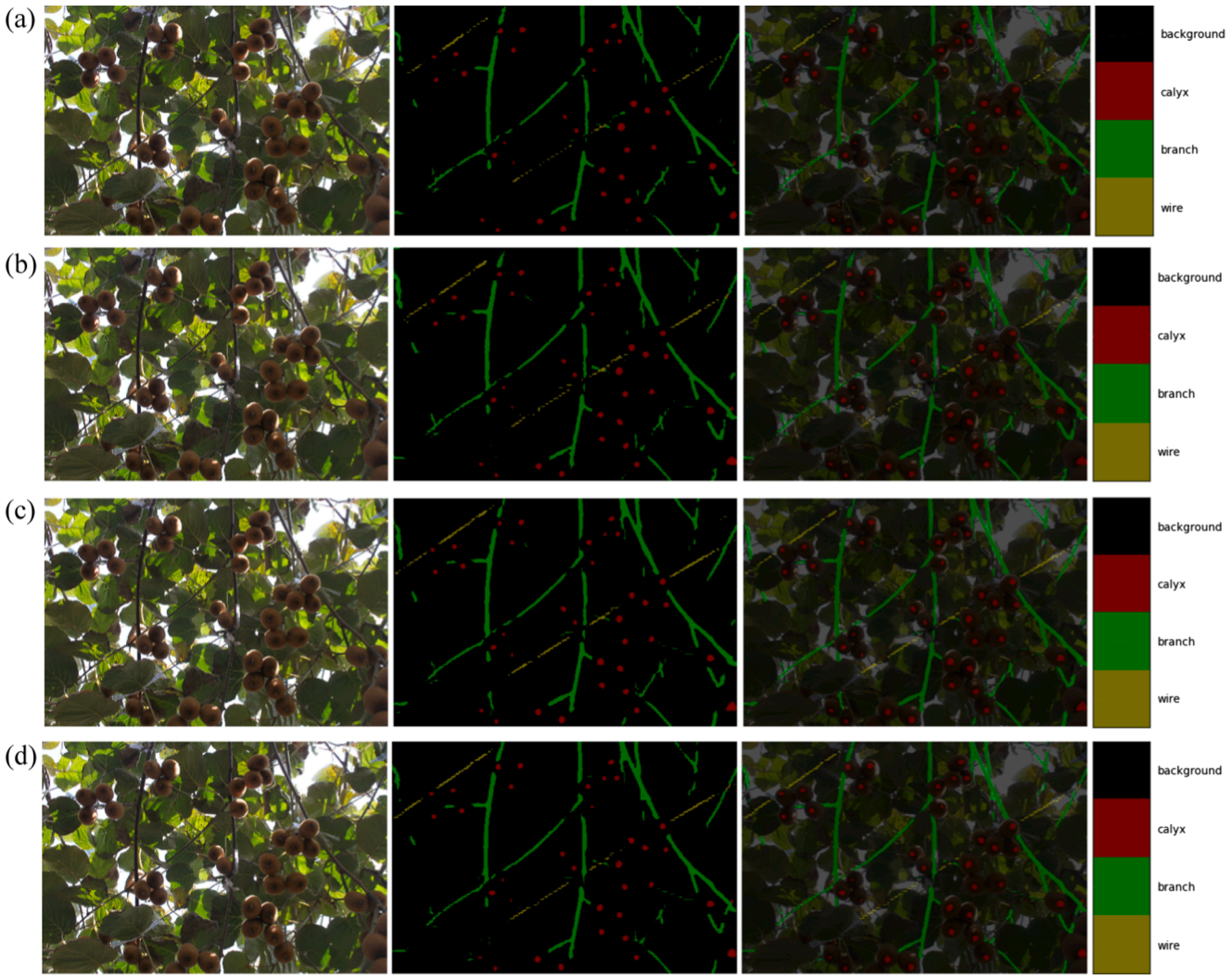


Fig. 7. Demonstrations of the segmentation maps and overlaps of DeepLabV3+ based on four backbones of Xception-71 (a), Xception-65 (b), ResNet-101 (c), and ResNet-50 (d) with the uniform weights. The black, red, green, and yellow represented the class of background, calyx, branch, and wire, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

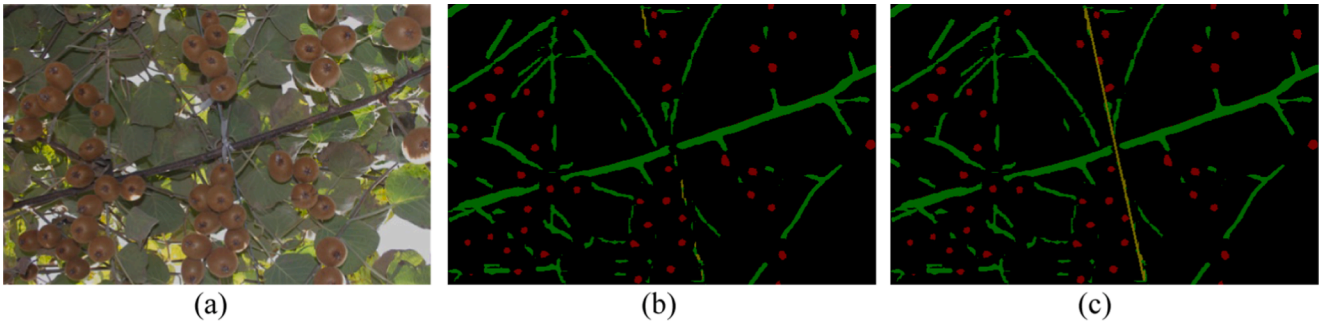


Fig. 8. Example of original kiwifruit canopy image (a), its segmentation map (b), and corresponding wire reconstruction image (c).

Table 5
Results of wire detection.

CDR (%)	MDR (%)	IDR (%)	Speed (ms/image)
92.4	6.1	1.5	6.4 ± 0.5

reducing the overall complexity and cost will be the focus of our future research. Others, additional development to segment calyxes, branches, and wires on night kiwifruit canopy image will help to improve the work efficiency of the robot.

CRediT authorship contribution statement

Zhenzhen Song: Data curation, Investigation, Writing - original

draft. **Zhongxian Zhou:** Investigation, Methodology, Writing - review & editing. **Wenqi Wang:** Methodology, Writing - review & editing. **Fangfang Gao:** Writing - review & editing. **Longsheng Fu:** Conceptualization, Data curation, Methodology, Supervision, Writing - review & editing. **Rui Li:** Methodology, Writing - review & editing. **Yongjie Cui:** 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.

Acknowledgement

This work was supported by the China Postdoctoral Science Foundation funded project (2019M663832); Fundamental Research Funds for the Central Universities of China (2452020170); National Natural Science Foundation of China (grant number 31971805); Recruitment Program of High-End Foreign Experts of the State Administration of Foreign Experts Affairs, Ministry of Science and Technology, China (grant number G20200027075).

References

Ayhan, B., Kwan, C., 2020. Tree, shrub, and grass classification using only RGB images. Remote Sens. 12, 1333. https://doi.org/10.3390/rs12081333.

Chang, J.W., Kang, S.J., 2018. Real-time vehicle detection and tracking algorithm for forward vehicle collision warning. J. Semicond. Technol. Sci. 18, 547–559. https://doi.org/10.5573/JSTS.2018.18.5.547.

Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation, In: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). Springer Verlag, Munich, pp. 833–851. DOI: 10.1007/978-3-030-01234-2_49.

Chen, W., Chen, S., Guo, H., Ni, X., 2020. Welding flame detection based on color recognition and progressive probabilistic Hough transform. Concurr. Comput. Pr. Exp. e5815 https://doi.org/10.1002/cpe.5815.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions, In: IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017. IEEE, Honolulu, HI, pp. 1800–1807. DOI: 10.4271/2014-01-0975.

Dong, R., Pan, X., Li, F., 2019. DenseU-Net-Based semantic segmentation of small objects in urban remote sensing images. IEEE Access 7, 65347–65356. https://doi.org/10.1109/ACCESS.2019.2917952.

Feng, A., Zhou, J., Vories, E., Sudduth, K.A., 2020. Evaluation of cotton emergence using uav-based narrow-band spectral imagery with customized image alignment and stitching algorithms. Remote Sens. 12, 1764. https://doi.org/10.3390/rs12111764.

Fu, L., Feng, Y., Wu, J., Liu, Z., Gao, F., Majeed, Y., Al-Mallahi, A., Zhang, Q., Li, R., Cui, Y., 2020a. Fast and accurate detection of kiwifruit in orchard using improved YOLOv3-tiny model. Precis. Agric. https://doi.org/10.1007/s11119-020-09754-y.

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

Fu, L., Gao, F., Wu, J., Li, R., Karkee, M., Zhang, Q., 2020c. Application of consumer RGB-D cameras for fruit detection and localization in field: A critical review. Comput. Electron. Agric. 177, 105687. https://doi.org/10.1016/j.compag.2020.105687.

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.

Fu, L., Wang, B., Cui, Y., Su, S., Gejima, Y., Kobayashi, T., 2015. Kiwifruit recognition at nighttime using artificial lighting based on machine vision. Int. J. Agric. Biol. Eng. 8, 52–59. https://doi.org/10.3965/j.ijabe.20150804.1576.

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. Comput. Electron. Agric. 176, 105634. https://doi.org/10.1016/j.compag.2020.105634.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, In: IEEE Conf. Comput. Vis. Pattern Recognit. pp. 770–778. DOI: 10.1109/CVPR.2016.90.

Ji, W., Meng, X., Qian, Z., Xu, B., Zhao, D., 2017. Branch localization method based on the skeleton feature extraction and stereo matching for apple harvesting robot. Int. J. Adv. Robot. Syst. 14, 1–9. https://doi.org/10.1177/1729881417705276.

Kang, H., Chen, C., 2020. Fruit detection, segmentation and 3D visualisation of environments in apple orchards. Comput. Electron. Agric. 171, 105302. https://doi.org/10.1016/j.compag.2020.105302.

Keskar, N.S., Nocedal, J., Tang, P.T.P., Mudigere, D., Smelyanskiy, M., 2017. On large-batch training for deep learning: Generalization gap and sharp minima, In: Int. Conf. Learn. Represent. ICLR, Toulon.

Liang, C., Xiong, J., Zheng, Z., Zhong, Z., Li, Z., Chen, S., Yang, Z., 2020. A visual detection method for nighttime litchi fruits and fruiting stems. Comput. Electron. Agric. 169, 105192. https://doi.org/10.1016/j.compag.2019.105192.

Lin, G., Tang, Y., Zou, X., Xiong, J., Li, J., 2019. Guava detection and pose estimation using a low-cost RGB-D sensor in the field. Sensors 19, 428. https://doi.org/10.3390/s19020428.

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.

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.

Majeed, Y., Karkee, M., Zhang, Q., Fu, L., Whiting, M.D., 2020a. Determining grapevine cordon shape for automated green shoot thinning using semantic segmentation-based deep learning networks. Comput. Electron. Agric. 171, 105308. https://doi.org/10.1016/j.compag.2020.105308.

Majeed, Y., Zhang, J., Zhang, X., Fu, L., Karkee, M., Zhang, Q., Whiting, M.D., 2020b. 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.

Marzougui, M., Alasiry, A., Kortli, Y., Baili, J., 2020. A lane tracking method based on progressive probabilistic hough transform. IEEE Access 8, 84893–84905. https://doi.org/10.1109/ACCESS.2020.2991930.

Mu, L., Cui, G., Liu, Y., Cui, Y., Fu, L., Gejima, Y., 2020. Design and simulation of an integrated end-effector for picking kiwifruit by robot. Inf. Process. Agric. 7, 58–71. https://doi.org/10.1016/j.inpa.2019.05.004.

Santos, T.T., de Souza, L.L., dos Santos, A.A., Avila, S., 2020. Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. Comput. Electron. Agric. 170, 105247. https://doi.org/10.1016/j.compag.2020.105247.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the Inception architecture for computer vision, In: IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2016. IEEE, Las Vegas, NV, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308.

UN Food & Agriculture Organization, 2020. Production of kiwi (fruit) by countries. Retrieved 2020-06-25.

Viola, P., Jones, M., 2001. Rapid object detection using a boosted cascade of simple features, In: Proc. 2001 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition. CVPR 2001. IEEE, Kauai, pp. 1193–1197. DOI: 10.1109/CVPR.2001.990517.

Williams, H.A., Ting, C., Nejati, M., Jones, M.H., Penhall, N.D., Lim, J.Y., Seabright, M.J., Bell, J., Ahn, H.S., Scarfe, A.J., Duke, M.D., MacDonald, B.A., 2020. Improvements to and large-scale evaluation of a robotic kiwifruit harvester. J. F. Robot. 37, 187–201. https://doi.org/10.1002/rob.21890.

Williams, H.A., Jones, M.H., Nejati, M., Seabright, M.J., Bell, J., Penhall, N.D., Barnett, J. J., Duke, M.D., Scarfe, A.J., Ahn, H.S., Lim, J.Y., MacDonald, B.A., 2019. Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms. Biosyst. Eng. 181, 140–156. https://doi.org/10.1016/j.biosystemseng.2019.03.007.

Yu, Y., Zhang, K., Yang, L., Zhang, D., 2019. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. Comput. Electron. Agric. 163, 104846. https://doi.org/10.1016/j.compag.2019.06.001.

Zhang, S., Ma, Z., Zhang, G., Lei, T., Zhang, R., Cui, Y., 2020a. Semantic image segmentation with deep convolutional neural networks and quick shift. Symmetry (Basel). 12, 427. https://doi.org/10.3390/sym12030427.

Zhang, Y. (Eds.), 2019a. Production of Fruit. Shaanxi Statistical Yearbook, 227-265.

Zhang, X., Fu, L., Karkee, M., Whiting, M.D., Zhang, Q., 2019b. Canopy segmentation using ResNet for mechanical harvesting of apples. IFAC-PapersOnLine 52, 300–305. https://doi.org/10.1016/j.ifacol.2019.12.550.

Zhang, Z., Igathinathane, C., Li, J., Cen, H., Lu, Y., Flores, P., 2020b. Technology progress in mechanical harvest of fresh market apples. Comput. Electron. Agric. 175, 105606. https://doi.org/10.1016/j.compag.2020.105606.

Zhou, J., He, L., Karkee, M., Zhang, Q., 2016. Effect of catching surface and tilt angle on bruise damage of sweet cherry due to mechanical impact. Comput. Electron. Agric. 121, 282–289. https://doi.org/10.1016/j.compag.2016.01.004.

Zhou, Z., Song, Z., Fu, L., Gao, F., Li, R., Cui, Y., 2020. Real-time kiwifruit detection in orchard using deep learning on Android™ smartphones for yield estimation. Comput. Electron. Agric. 179, 105856. https://doi.org/10.1016/j.compag.2020.105856.