Computers and Electronics in Agriculture 209 (2023) 107840

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Thin wire segmentation and reconstruction based on a novel image overlap-partitioning and stitching algorithm in apple fruiting wall architecture for robotic picking

Hanhui Jiang [a], Xiaoming Sun [a], Wentai Fang [a], Longsheng Fu [a,b,c,d,*], Rui Li [a], Fernando Auat Cheein [e,*], Yaqoob Majeed [f]

[a] College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling, Shaanxi 712100, China
[b] Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture and Rural Affairs, Yangling, Shaanxi 712100, China
[c] Shaanxi Key Laboratory of Agricultural Information Perception and Intelligent Service, Yangling, Shaanxi 712100, China
[d] Northwest A&F University Shenzhen Research Institute, Shenzhen, Guangdong 518000, China
[e] UK National Robotarium, Edinburgh Centre for Robotics, School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh, UK
[f] Faculty of Agricultural Engineering and Technology, University of Agriculture, Faisalabad 38000, Pakistan

ARTICLE INFO

Keywords:
BlendMask
Feature point extraction and fitting
Fruiting wall architecture
Image overlap-partitioning and stitching
Wire reconstruction

ABSTRACT

The layout of orchards usually requires the use of wires, to provide sturdy support. Such is the case of apple trees in fruiting wall architecture, where wires are conducive for mechanical harvesting and especially robotic picking. However, wires may cause damage to the robotic gripper, especially when direct picking occluded apple fruits. Hence, the importance of identifying the wires of a fruit wall architecture. In this study, a pixel-level segmentation network, BlendMask, was adopted to segment wires. The wires are thin and normally behind the branches or leaves, making difficult for their identification. Therefore, a novel data processing algorithm called image overlap-partitioning and stitching (IOS) is proposed for BlendMask to segment the wires. A total of 82 RGB (Red, Green, and Blue) images registered to create a raw dataset. The dataset was augmented and then overlap-partitioned into 12,736 images with a resolution of 800 × 1024 pixels and corresponding annotation files based on input size of BlendMask to make an overlap-partitioned dataset. Then BlendMask was trained with the overlap-partitioned dataset and tested on the other images, where additional stitching for the overlap-partitioned images was needed in the image testing dataset. Results showed that BlendMask with IOS obtained Intersection over Union (IoU) and pixel accuracy of 43.86 % and 61.01 %, respectively. It achieved a better average precision of 38.75 % with IoU of 0.5 on the overlap-partitioned dataset, which was 38.42 % higher than the full image dataset. Moreover, a reconstruction method based on feature point extraction and fitting was proposed to estimate wire skeletons, which achieved a reconstruction accuracy of 90.70 %. These results showed a promising potential using segmentation and reconstruction methods for identifying wires and thus providing a basis for robotic picking in modern orchards.

1. Introduction

Apple is the third most popular fruit in the world based on production quantities. The estimated production number of apples in the world was 7.1 × 10^7 tons in 2009 and 8.7 × 10^7 tons in 2019, which have increased 1.6 × 10^7 tons for a decade (UN Food Agriculture Organization, 2022). Currently, apple harvesting highly relies on human labor force while facing the growing production number of apples (Zhao et al., 2011; Fu et al., 2020; Zhang et al., 2016; Arikapudi and Vougioukas, 2021). Labor costs for apple harvesting accounted for more than 30 % of production costs (He et al., 2019; Hu et al., 2020; Chen et al., 2023). Mechanical harvesting not only enables picking efficiency but also has the potential to reduce the labor cost, which is expected to become an alternative option to manual picking.

Wires provide sturdy support to apple trees in modern orchards, which are conducive for mechanical harvesting and especially robotic picking. Traditional multi-dimensional architectures have developed into modern orchards (Hu et al., 2020). Wires have been widely used in

* Corresponding author.
E-mail addresses: fulsh@nwafu.edu.cn, longsheng.fu@outlook.com (L. Fu), F.Auat@hw.ac.uk (F.A. Cheein).

https://doi.org/10.1016/j.compag.2023.107840
Received 17 September 2022; Received in revised form 28 February 2023; Accepted 9 April 2023
Available online 20 April 2023
0168-1699/© 2023 Elsevier B.V. All rights reserved.

modern orchards for fruiting walls' building. For apple trees in fruiting wall architecture, the wires are employed to alleviate the pressure of the trees to make accessible for robotic picking (Majeed et al., 2020b). Specifically, the wires are tied with a spacing horizontally to branches, where normally 6 or 7 wires are needed to support a row of trees, as shown in Fig. 1. Following, a pipe is tied to the bottom wire for irrigation, which is significantly thicker than wire and intertwine with the bottom wire.

Accurate and robust identification of the wires is essential for robotic picking. Apples are commonly occluded by obstacles. Most of studies focused on identifying occluded apple fruits successfully, but ignored obstacles occluding apple fruits (Fu et al., 2020; Gao et al., 2020; Gené-Mola et al., 2021; Hameed et al., 2022; Jia et al., 2021b; Niu et al., 2017; Silwal et al., 2017; Wan and Goudos, 2020; Wang and He, 2019; Zhao et al., 2021; Wu et al., 2022b; Tang et al., 2023), where specific information of obstacles not only is beneficial to decide higher-level picking strategies for the occluded apple fruits, but also improves the harvesting efficiency of picking robots (Jia et al., 2020; Wu et al., 2022a). Hence, the need to have a precise approach to identify the wires on the field.

Deep learning algorithms have been widely applied to segment orchard images, which is a primary step in object identification (Fu et al., 2019). For example, Jia et al. (2021a) improved Segmenting Objects by Locations (SOLO) to segment green fruits in the natural orchard, which reached a precision (P) of 0.96. Zhang et al. (2021) used DeepLabV3 + with ResNet-18 to segment four classes, i.e., branches, trunks, apples, and leaves, which achieved a mean per-class P of 0.97. Long et al. (2021) improved Mask Region-based Convolutional Neural Network (Mask R-CNN) to segment tomato fruit and obtained the P of 0.95. Mask R-CNN was applied for identifying citrus branches and apple fruits, which reported the P of 0.98 and 0.86, respectively (Gené-Mola et al., 2020; Yang et al., 2019). Xi et al. (2021) compared BlendMask and Mask R-CNN in segmenting ginkgo tree crowns of unmanned aerial vehicle (UAV) multispectral images, where BlendMask obtained better values with 0.95 in mean average precision based on intersection over union (IoU) of 0.5. Those studies based on deep learning algorithms have obtained acceptable results on orchard image segmentation.

Wires have been too thin to be robustly segmented compared to other obstacles on the orchard images, according to up to date literature review. Majeed et al. (2020b) applied convolutional neural network (CNN)-based segmentation network (SegNet) to segment trunks, branches, and wires on apple tree images, which reached IoUs of 0.61, 0.46, and 0.29, respectively. Song et al. (2021) employed DeepLabV3 + with ResNet-101 to segment calyxes, branches, and supporting wires on kiwifruit images, which achieved IoUs of 0.68, 0.70, and 0.42, respectively. The wires had lower IoUs compared with other obstacles on the orchard images (Majeed et al., 2020b; Song et al., 2021). One possible reason is that wires in orchard are too thin to be easily segmented from a far view image, where the proportion of wire pixels in an image (0.6 %) was much smaller than that of background (89.3 %) and branches (9.0 %) (Song et al., 2021). Increasing the proportion of the wire pixels in the images seems to segment wires more robustly compared with maintaining original proportion of the wire pixels in the images to segment the wires.

Reconstructing the wires based on their agronomic usage is another possible solution to estimate wire skeletons. During harvesting season, the wires are normally behind the branches or leaves, and thus partly invisible on the orchard images (Suo et al., 2021). Invisible wires are difficult to be segmented but may be reconstructed from the visible wires based on their agronomic properties. Polynomial fitting has been commonly used to reconstruct thin branches on the orchard images (Amatya et al., 2016; Majeed et al., 2020a; Wan et al., 2022; Zhang et al., 2018), while wires are not only thin but are shown with a spacing horizontally on the orchard images. Therefore, further research on reconstructing the wires can be suggested based on polynomial fitting.

To address the above-mentioned problem, a segmentation method and a reconstruction method are proposed herein, to identify the wires in the natural orchard. Images acquired from the natural orchard are overlap-partitioned and then fed to BlendMask, a pixel-level

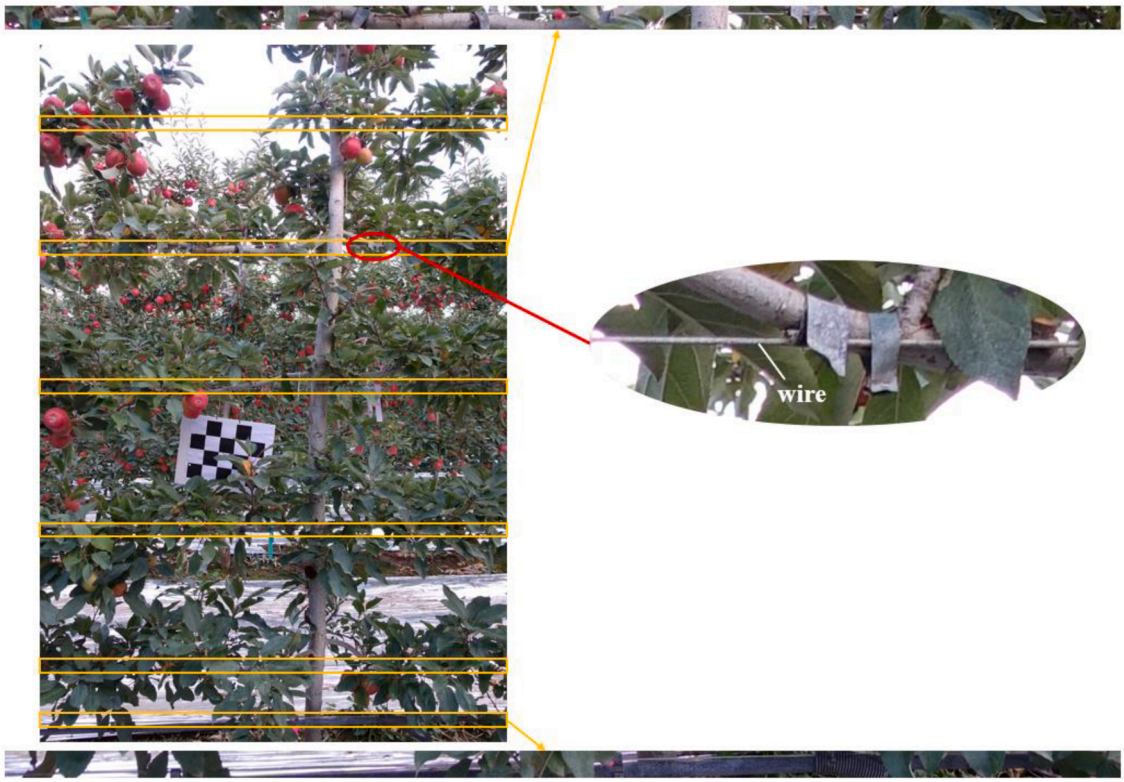


Fig. 1. An example image showing spatial location of the wires. Orange rectangles artificially drew on the image to show the location of the wires. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

segmentation network, whereupon the thin wires on the images segmented in pixel level are reconstructed using polynomial fitting. The remaining of the work is organized as follows. In Section 2, the materials and methods are described in terms of image processing and specific method. In Section 3, the results are presented and discussed. Lastly, in Section 4, the conclusions and prospects of this paper are described.

2. Materials and methods

This work is focused on identifying the wires in fruiting wall architecture using segmentation and reconstruction methods. The specific operation steps in this work are shown in Fig. 2. Briefly, A novel data processing algorithm called image overlap-partitioning and stitching (IOS) is proposed for BlendMask to segment wires. A total of 82 RGB (Red, Green, and Blue) images were acquired in a commercial orchard of Prosser, Washington state, USA. The images were annotated manually with polygons and then saved as JSON format to make a raw dataset. The raw dataset was augmented and then overlap-partitioned into 12,736 images with resolution of 800 × 1024 pixels with corresponding annotation files based on input size of BlendMask, and finally cleaned by deleting images without corresponding annotation files to make an overlap-partitioned dataset. BlendMask was trained with the overlap-partitioned dataset and tested on overlap-partitioned images, where the overlap-partitioned images were additionally stitched in image testing. The wires on the stitched images were reconstructed using a method based on feature point extraction and fitting (FPEF) proposed in this work. Additionally, performance of wire segmentation and reconstruction are also evaluated in this work. Detailed explanations of the methods are provided in the following sections.

2.1. Image acquisition

Images were captured during a harvesting season during 2017, in a commercial orchard of Prosser, Washington State, USA. The Scifresh apple trees were around 4.0 m tall and trained in seven layers where the branches were tied to wires at each layer. A schematic for image acquisition is illustrated in Fig. 3. The distance between tree rows (D1) was about 2.6 m. A camera (SONY RX100M4, Tokyo, Japan) was mounted on a tripod to maintain a vertical height (H) of about 1.4 m from the ground. The distance between the camera and apple fruit tree (D2) was about 2.5 m. In total, 82 images were collected by the camera (9 mm focal length of lens, 52° horizontal field of view, 73° vertical field of view, and 83° display field of view) with 3648 × 5472 pixels resolution in both daytime (sunny days) and nighttime (with artificial light from camera's flash) and then saved in JPEG format.

2.2. Image annotation

The annotation to the full image can connect more contextual information and thus make labor force easier to annotate compared with the annotation to the overlap-partitioned images. All full images were annotated using Labelme software (an image labeler toolbox developed by MIT's Computer Science and Artificial Intelligence Laboratory). Specifically, wires at each layer on the images were discrete and thus labeled using polygons manually in pixel level, where invisible wires were not labeled. Examples of an RGB image and its corresponding annotated image are shown in Fig. 4a and Fig. 4b, respectively. Different polygons were distinguished with different colors and saved as point sets in annotation files. A raw dataset was generated based on annotated images and was randomly divided into training (68) and testing (14) sets at a ratio close to 5:1.

2.3. Dataset augmentation

The training set was not enough to train BlendMask model, and therefore it was augmented. Small number of training set may cause overfitting or nonconvergence of training BlendMask. To address this issue, dataset augmentation is an effective way to improve sufficiency and diversity of training set. Specific methods for dataset augmentation adopted in this study were horizontal mirroring, brightness variation, contrast variation, and saturation variation using Imgaug (a library for image augmentation in machine learning experiments, available on https://imgaug.readthedocs.io/en/latest/index.html). The image mirroring was carried out using function 'Fliplr'. Function'MultiplyBrightness', 'LinearContrast', and 'MultiplySaturation' were adopted for brightness variation, contrast variation, and saturation variation, respectively, where a proportional coefficient of 0.9 was fed to adjust brightness, contrast, and saturation of the images. The raw training set was finally augmented to 272 images by the above four mentioned methods. Thus, there were total 286 images in the augmented dataset.

2.4. Dataset overlap-partitioning

As mentioned earlier, wires are too thin to be easily segmented on the images acquired from the orchard. To address this issue, the dataset was overlap-partitioned to increase the proportion of wire pixels in the images. Graphic of overlap-partitioning an image is shown in Fig. 5. A raw image with the resolution of 3648 × 5472 pixels was overlap-partitioned to 56 images with the resolution of 800 × 1024 pixels from top-left to bottom-right based on input size of BlendMask, where the overlap-partitioned images overlapped each other by 20 %. Besides, empty space of the overlap-partitioned images in last row or last column was padded with green color. The number of images in the dataset after overlap-partitioning was 16016, while 15,232 images in the training set. The dataset is available on https://github.com/fu3lab/Apple_wire_RGB_images.

Annotation files were overlap-partitioned along with images. The process of overlap-partitioning an annotation file is shown in Fig. 6. Wires on the images were labeled using polygons and saved in the annotation files as point sets. Therefore, the point sets were manipulated to generate overlap-partitioned annotation files. Originally, coordinates of points in the point sets were their distance from a reference point located at top-left in the raw image. The coordinates of points were transformed when the location of the reference point was changed to top-left in overlap-partitioned images. Then, if all points of the point sets were out of the overlap-partitioned images, they were thus deleted,

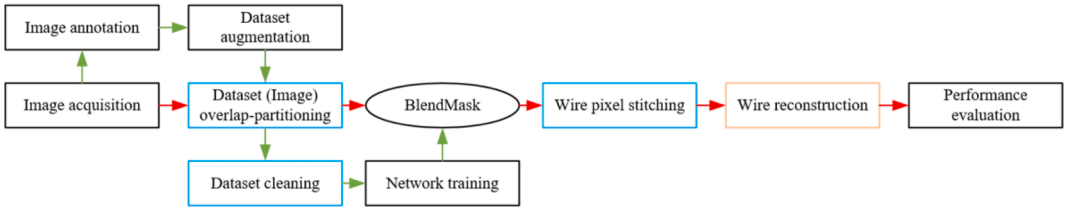


Fig. 2. Overall schematic flow chart. Rectangles represent operations of this work while the ellipse represents BlendMask model used for training and testing. Specifically, blue and orange rectangles represent operations of IOS and FPEF, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

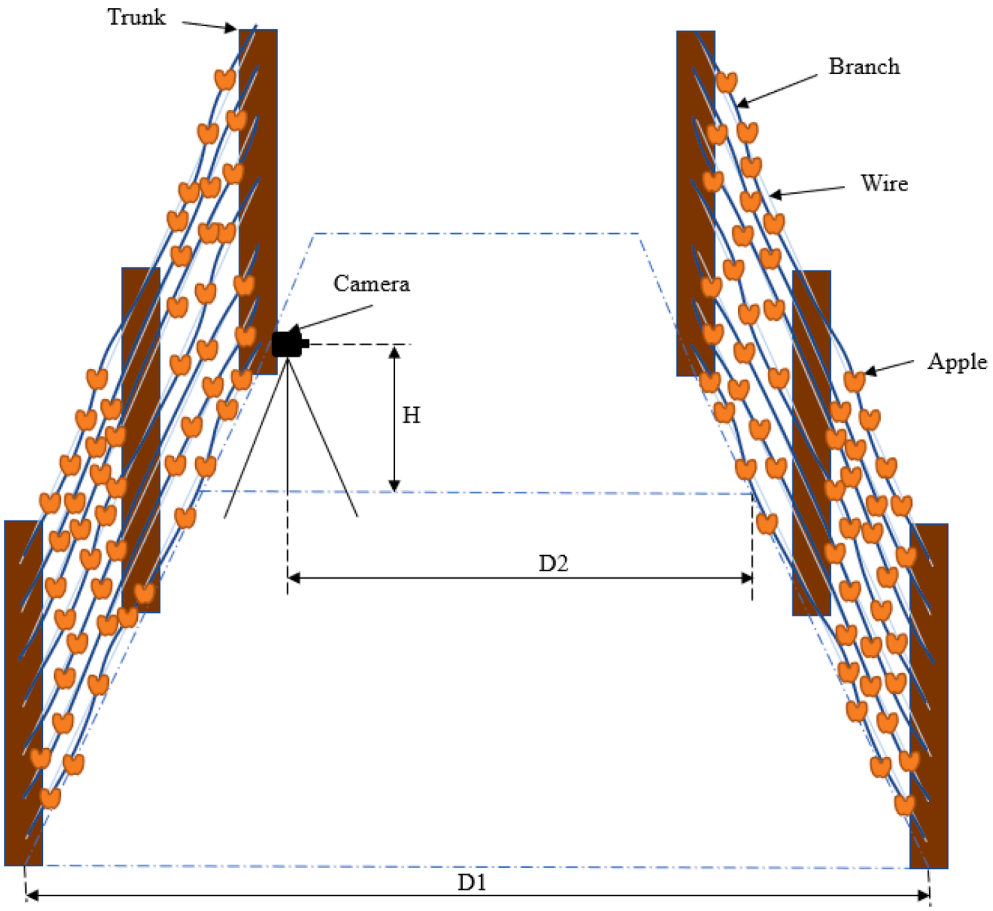


Fig. 3. A schematic for image acquisition.

otherwise, they were thus saved in the corresponding overlap-partitioned annotation files if the all points of the point sets were in the overlap-partitioned images. If the points of the point sets were not all in the overlap-partitioned images, the points out of the overlap-partitioned images were deleted while boundary points that were defined as intersection between borders of the overlap-partitioned images and the polygons were increased. Either the points of point sets in the overlap-partitioned images or boundary points were both sorted with counterclockwise or clockwise direction (counterclockwise in this work). Finally, 56 overlap-partitioned annotation files were saved and generated from a raw annotation file.

2.5. Dataset cleaning

After overlap-partitioning, some of images in the training set had no annotations and thus were deleted. If these images were input in BlendMask, no target was trained, which produced an increasing number of negative samples. On the one hand, the negative samples could not provide useful learning information for training BlendMask, resulting in a waste of training source. On the other hand, data imbalance brought by the negative samples affected detrimentally the performance of the model (Mellor et al., 2015). To solve this problem, images without annotation in the training set were deleted through the program and not participated in training. The number of images in the training set was 11,952 (3280 was deleted), which together with images in the testing set to make a final dataset (12736) for training and testing.

2.6. Network training

Training architecture for wire segmentation was based on BlendMask. Details of BlendMask architecture are shown in Fig. 7 (Chen et al., 2020). The specific steps of training were as the following. Transfer learning using pre-trained model and fine-tuning parameters was employed to learn quickly and achieve robust and accurate results in this work. ResNet-101 model trained on the ImageNet database (He et al., 2016), as the pre-trained model, were utilized to initialize of weights. Then the hyperparameters of BlendMask were selected in this work after five trials, as shown in Table 1. BlendMask model was trained on a desktop computer with AMD Ryzen 7 5800X 8-Core Processor (3.80 GHz) CPU, NVidia GeForce GTX 3080 Ti 12 GB GPU (10,240 CUDA cores), and 64 GB of memory based on Ubuntu 18.04 system. The experiments were implemented based on Adelaidet (an open source toolbox for multiple instance-level recognition tasks on top of Detectron2).

2.7. Wire pixel stitching

Wire pixels were segmented on the overlap-partitioned images and thus needed to be stitched to obtain complete segmentation results. The stitching process for wire pixels was illustrated based on two adjacent overlap-partitioned images in a same row, as shown in Fig. 8. Segmented wire pixels on the two adjacent overlap-partitioned images were labeled using red and yellow lines, respectively (see Fig. 8a). As mentioned in Section 2.4, the reference points were located at top-left in the overlap-partitioned images were denoted as (0, 0) and transformed based on relative locations from the overlap-partitioned images to the raw image (see Fig. 8b), which correspondingly transformed the coordinates of segmented wire pixels. Specifically, the coordinates of segmented wire pixels after coordinate transformation were derived from that before coordinate transformation, which defined in Eq. (1), Eq. (2), Eq. (3) and



(a) (b)

Fig. 4. Examples of an RGB image in fruiting-wall architecture (a) and its annotated image using polygons (b). Different polygons were distinguished with different colors.



Fig. 5. A graphic of overlap-partitioning a raw image, showing an example of overlap-partitioned images arranged in rows and columns of the raw image. The images with red borders overlapped the images with yellow borders by 20%. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Eq. (4), respectively. The coordinates of the transformed segmented wire pixels were then fused with each other, which were labeled using purple lines (Fig. 8c). The fusion of transformed segmented wire pixels is based on "OR" operation: If a pixel was classified as wire in at least one of the two images, it was labeled as wire in the stitched image, using a decision variable that defined in Eq. (5).

$$X_{i,\,j} = A \times (j-1) + x_{i,\,j} \tag{1}$$

$$Y_{i,\,j} = B \times (i-1) + y_{i,\,j} \tag{2}$$

$$A = \frac{1-r}{1+r} \times w \tag{3}$$

$$B = \frac{1-r}{1+r} \times h \tag{4}$$

where $x_{i,j}$ and $y_{i,j}$ are coordinates of segmented wire pixels before coordinate transformation; $X_{i,j}$ and $Y_{i,j}$ are coordinates of segmented wire pixels after coordinate transformation; i and j represent row and column numbers of overlap-partitioned images in the raw image; A and B are location parameters (533 and 683 in this work); w and h are input size of the network (800 and 1024 in this work) and r is overlapping rate of the adjacent images (20 % in this work).

$$P(X^{`}, Y^{`}) = \begin{cases} 1 & P\left(X_{i,\,j},\ Y_{i,\,j}\right) = 1 \text{ or } P\left(X_{i,\,j+1},\ Y_{i,\ \ j+1}\right) = 1 \\ 0 & P\left(X_{i,\,j}, Y_{i,\,j}\right) = 0 \text{ and } P\left(X_{i,\,j+1}, Y_{i,\,j+1}\right) = 0 \end{cases} \tag{5}$$

where $P(X, Y)$ is a decision variable with wire pixels in coordinate (X, Y) if the value is one and instead not wire pixels in coordinate (X, Y) if the value is zero; $X^{`}$ and $Y^{`}$ are coordinates of stitched images.

2.8. Wire reconstruction

Wires were required to be reconstructed based on their agronomic

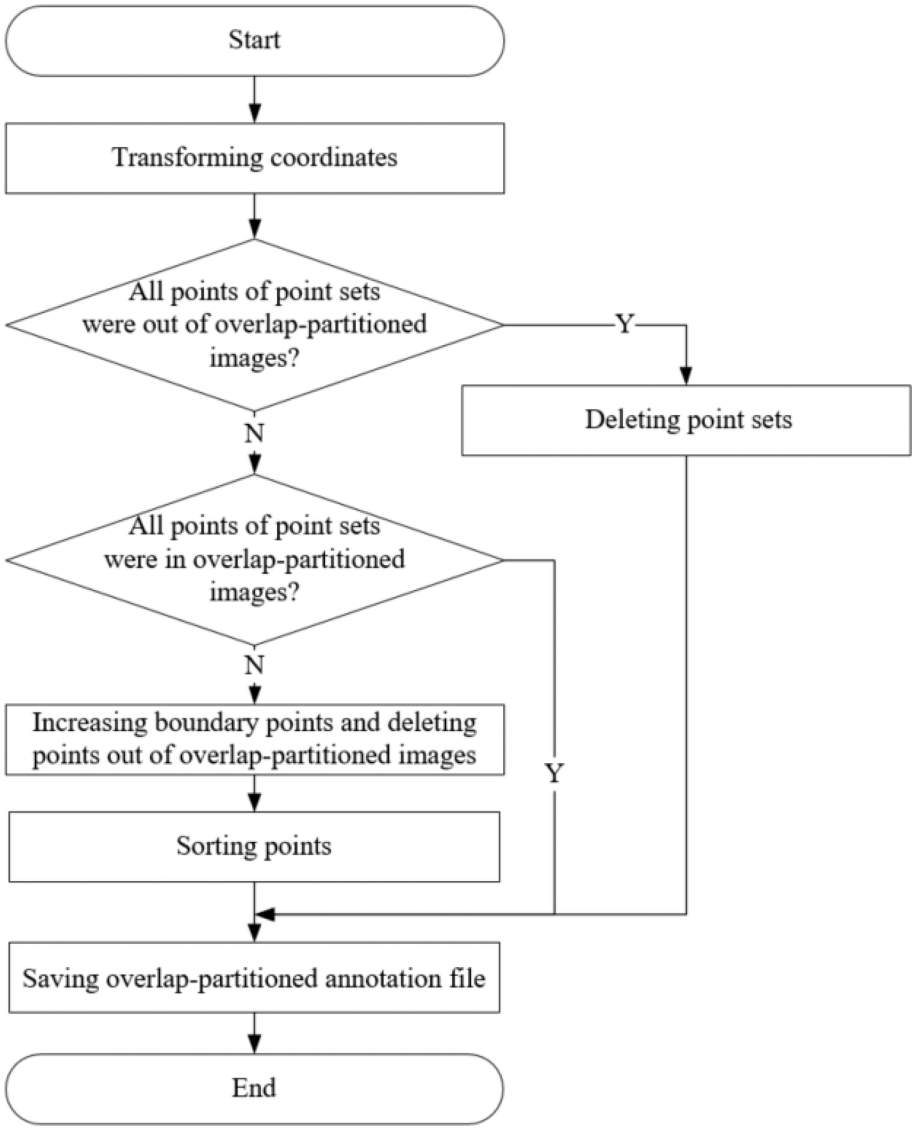


Fig. 6. The process of overlap-partitioning an annotation file.

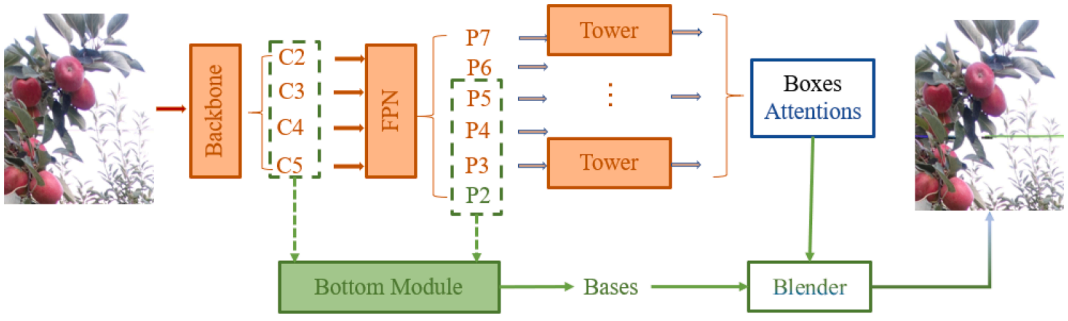


Fig. 7. Architecture of BlendMask.

properties to estimate wire skeletons. The wires are thin and maintained at a certain distance between them except for the bottom wire, which can be used as constraint conditions for wire reconstruction. Hence, FPEF adopted feature points extracted from the stitched segmented wire pixels with constraints for fitting, as shown in Fig. 9. The polygons representing the stitched segmented wire pixels were clustered in y-axis and thus belonged to same wires when maximum distances between them were less than the clustering threshold that was set to 110 pixels. Then, wires tied to the tree branches are filtered if the distances between them are less than filter threshold. The filter threshold is set to 800 pixels to limit the minimum distance between the wires tied to the tree branches depending on the wire agronomic use. Feature points were extracted from the wires and corresponding clustered polygons, which included polygon endpoints and wire restriction points, respectively. The polygon endpoints were minimum and maximum points in x-axis of the clustered polygons while wire restriction points were two midpoints

Table 1
Training parameters of BlendMask.

Training parameters	Values
Backbone	ResNet-101
Base learning rate	0.0001
Warm-up iteration	1000
Weight decay	0.0001
Learning rate decay steps	(280000, 360000)
Learning rate reduced factor	10
Train iteration	400,000
Images per batch	2
Momentum	0.9
Mask format	Polygon

from shorter sides of minimum circumscribed rectangle for the wires. Finally, the feature points were fed to one-order polynomial equation for fitting wire skeletons, defined in Eq. (6).

$$\widehat{f} = a_1 \times t + a_0 \tag{6}$$

$$E = \sum_{i=1}^{m} (\widehat{f}_i - f_i)^2 \tag{7}$$

where a_0 and a_1 are parameters of one-order polynomial equation, making a minimal deviation of Eq. (7), which m is the number of feature points and f_i and $\widehat{f}_i$ are ground truth and fitted values of feature point i.

It is worth mentioning that other approaches have been used in the literature to detect and characterize wires in other contexts, with non-concluding results. For example, the progressive probabilistic Hough transform (PPHT, implemented in this work for comparison purposes). It is a probabilistic algorithm that uses a random sampling of the edge points, which is commonly used for line detection and applied to be compared with polynomial fitting for wire estimation in this work (Marzougui et al., 2020). Other operations (Polygon cluster and wire filter) were as same in the FPEF except using PPHT. Edge points were input in PPHT instead feature points were extracted for polynomial fitting. The whole experiment was carried out using the OpenCV and Python language. Function "cv2.Canny" was used and then obtained edge points of wires, which was fed into Function "cv2.HoughLinesP" as input for wire estimation. After that, two coordinates of start and end points from detected lines returned by the function. The detected lines were filtered with the filter threshold and extended to the entire image based on the two coordinates.

2.9. Performance evaluation

Performance of BlendMask was evaluated using average precision (AP) and average precision with IoU = 50 % (AP50). Specifically, AP50 showed the average precision with IoU of 0.5 while AP showed a mean value with IoU from 0.5 to 0.95 with a step size of 0.05. AP and AP50 were calculated by precision (P) and recall (R), which was defined in Eq. (8) and Eq. (9).

$$\mathrm{AP} = \left(\sum_{IoU=0.5}^{0.95} \int_0^1 P_{IoU}(R) dR \right) \Big/ 10 \tag{8}$$

$$\mathrm{AP50} = \int_0^1 \mathrm{P}_{\mathrm{IoU}=0.5}(\mathrm{R}) \mathrm{dR} \tag{9}$$

where P was the number of correctly segmented wires divided by the sum of the number of correctly segmented wires and the number of wrong segmented wires, and R was the number of correctly segmented wires divided by the sum of the number of correctly segmented wires and the number of ground truth that were not segmented.

Performance of wire segmentation was evaluated using IoU and Pixel Accuracy (PA). IoU computed overlapping area between predicted wire pixels and ground truth wire pixels divided by the area of union between them, as shown in Eq. (10). PA simply found the ratio of correctly wire pixels divided by the total number of wire pixels, as defined in Eq. (11).

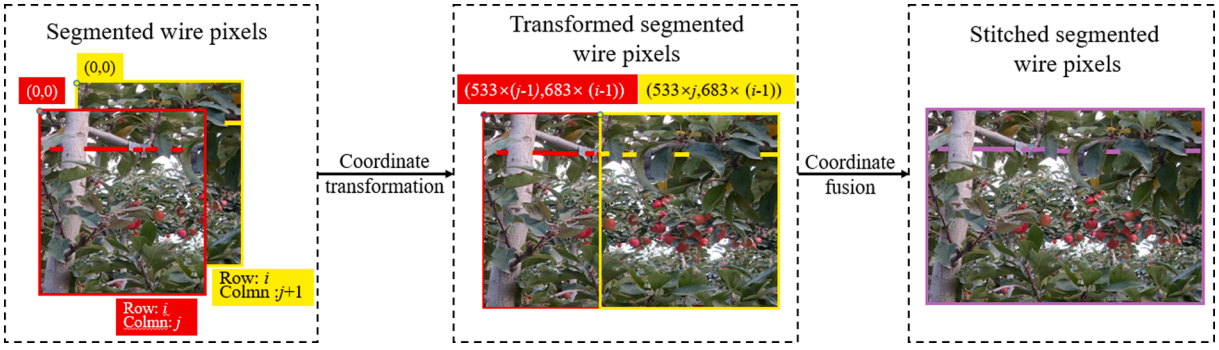


Fig. 8. The wire pixels stitching process. (a) Segmented wire pixels on the two adjacent overlap-partitioned images using red and yellow lines; (b) Transformed segmented wire pixels; (c) Stitched segmented wire pixels using purple lines. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

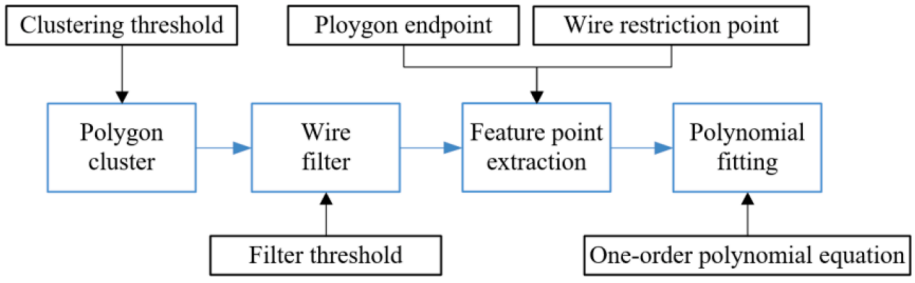


Fig. 9. The flowchart of FPEF. The blue boxes represented operations of FPEF while the black boxes represented key points to the corresponding steps. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{10}$$

$$PA = \frac{\sum_{i=0}^{K} p_{ii}}{\sum_{i=0}^{K} \sum_{j=0}^{K} p_{ij}} \tag{11}$$

where A and B denoted the sum of ground truth wire pixels and predicted wire pixels; K was the total number of categories (including background), and p_{ij} was the number of pixels of class i in ground truth images predicted as class j on segmented images, respectively.

Three metrics, reconstruction accuracy (RA), missed rate (MR), and incorrect rate (IR) were used to evaluate the performance of wire reconstruction. RA referred to the percentage of the number of correctly reconstructed wires to actual number of the reconstructed wires in the testing set. MR referred to the percentage of the number of missed reconstructed wires to actual number of the reconstructed wires in the testing set. IR referred to the percentage of the number of incorrect reconstructed wires to actual number of the reconstructed wires in the testing set. The RA, MR, and IR were calculated as Eq. (12), Eq. (13), and Eq. (14), respectively.

$$RA = CN / N_{all} \tag{12}$$

$$MR = MN / N_{all} \tag{13}$$

$$IR = IN / N_{all} \tag{14}$$

where correct number (CN) was the number of correctly reconstructed wires, missed number (MN) was the number of missed reconstructed wires, incorrect number (IN) was the number of incorrected reconstructed wires, and N_{all} was the actual number of wires, respectively.

3. Results and discussion

3.1. Performance of wire segmentation

BlendMask with IOS was used for wire segmentation and includes not only overlap-partitioning preprocess on full images before input to BlendMask model but stitching postprocess on the overlap-partitioned images after output by BlendMask model, which obtained wire segmentation results on the full images. Among them, BlendMask model used in BlendMask with IOS was trained by the overlap-partitioned dataset. Therefore, performance of BlendMask model with overlap-partitioned dataset and BlendMask with IOS were evaluated and discussed, respectively. Specific contents are as follow.

3.1.1. Performance of BlendMask models with overlap-partitioned dataset

The BlendMask model with overlap-partitioned dataset were compared with BlendMask model with full image dataset to validate the impact of overlap-partitioned dataset on performance of BlendMask. The full image dataset was made by image annotation and augmentation only (no dataset overlap-partitioning or dataset cleaning operation) and resized to 800 × 1024 resolution based on the aspect ratio of full images (lack of pixels is padded to zeros) before it was fed to BlendMask. The same training parameters as be trained overlap-partitioned dataset were set when BlendMask was trained by full image dataset.

Training loss curves of BlendMask models with full image (without dataset overlap-partitioning or dataset cleaning operation) and overlap-partitioned dataset (with dataset overlap-partitioning or dataset cleaning operation) are shown in Fig. 10. The training number of iterations for the two models was set as 400000. The loss values of the BlendMask models with overlap-partitioned and full image dataset decreased as the number of iterations increased and stabilized with the lowest values of 0.711 and 0.747 after 350,000 and 320,000 iterations, respectively. The BlendMask model with overlap-partitioned dataset achieved lower loss value but took longer to stabilize than that with full image dataset. The stable loss values demonstrated that the two models both trained appropriately with good convergence ability.

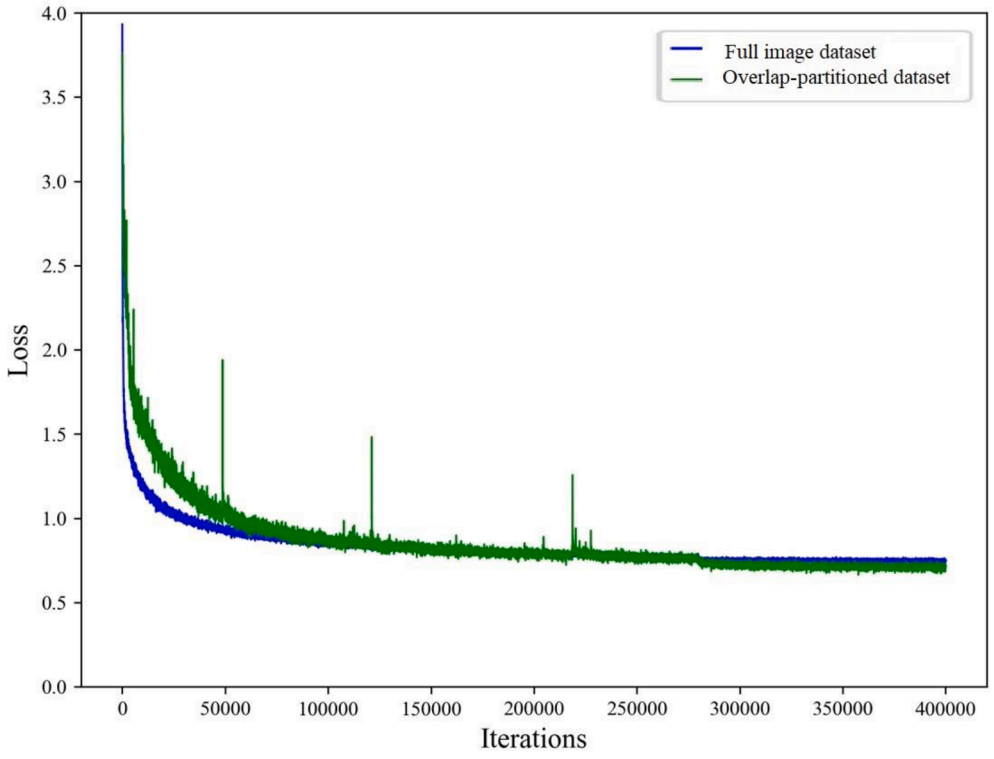


Fig. 10. Training loss curves of BlendMask model based on full image dataset and overlap-partitioned dataset, respectively.

The BlendMask model with overlap-partitioned dataset outperformed that with full image dataset. The models with overlap-partitioned and full image dataset obtained the AP of 13.42 % and 0.13 % and the AP50 of 38.75 % and 0.33 %, respectively. Apparently, it showed that BlendMask model was more sensitive to overlap-partitioned dataset than full image dataset. The reason was that the overlap-partitioned dataset had a higher proportion of wire pixels, which was easier to learn wire features compared with full image dataset. It can be concluded that the overlap-partitioning operation improved the generalization of the BlendMask model, but further conclusions are needed on wire segmentation.

Cropping operation of data augmentation in other studies is similar with overlap-partitioning operation of our algorithm, where they both increase not only the number of the images but the proportion of small object pixels in the images. However, the cropping operation of data augmentation in other studies sets "crop size" without deciding crop location and thus makes augmented images in same size but random contents in every training epoch, which cannot guarantee that raw wire data is fully used. The overlap-partitioning operation of our algorithm can ensure that the images are fully overlap-partitioned into different parts based on input size of BlendMask and overlap rate, not just a random part, which allows all wires in the images to be trained and is helpful to network completely and precisely learn the relevant features.

3.1.2. Performance of BlendMask with IOS on wire segmentation

The purpose of wire segmentation is to obtain wire segmentation results on the full images. Therefore, BlendMask with IOS is needed to be compared with BlendMask without IOS to evaluate wire segmentation results, where BlendMask without IOS directly input the full images to BlendMask with full image dataset without any postprocess.

Table 2 displays segmentation results of BlendMask with and without IOS. BlendMask with IOS obtained IoU and PA of 43.86 % and 61.01 %, which increased by 41.85 % and 58.89 % compared with that without IOS (IoU and PA of 2.01 % and 2.12 %), respectively. It is clear that IOS has greatly increased the IoU and PA. The cause was that IOS used BlendMask with an overlap-partitioned dataset that made it simpler to learn wire features and correctly stitched to wire pixels. Some insights from other studies were given when the wires segmented from different orchard environments were discussed along with the results achieved in this work. For the same variety, Majeed et al. (2020b) achieved IoU of 0.29 in segmenting thin wire on Scifresh apple trees during dormant season using SegNet on 1920 × 1080 pixels images. Song et al. (2021) employed DeepLabV3 + to segment wires in kiwifruit orchard and reported IoU of 0.42 for 66 images of 512 × 341 pixels resolution. Compared with above studies, images that was far-view captured during harvesting season in our study had little proportion of wire pixels and thus more difficult to be segmented, which was verified by IoU of 2.01 % using BlendMask without IOS in our work. However, the increase of IoU achieved by IOS suggested a valuable enhancement of the methodology and the results obtained from IOS indicated that it was potential to further estimate wire skeletons.

It took 0.13 ms and 54.69 ms to partition and stitch a raw image with a resolution of 3648 × 5472 pixels while taking 80.16 ms for an overlap-partitioned image with a resolution of 800 × 1024 pixels to be processed by BlendMask with IOS. The model processing speed for a raw image was thus 4488.96 ms with IOS, which was longer than without IOS for the raw image. The reason was that the raw image was overlap-partitioned with an overlapping rate and thus caused that the sum of pixels in the overlap-partitioned images was more than that in the raw image.

Nevertheless, the observation to segmented wire pixels on the resulting images presented more insights into the results. An example of a test image and its corresponding segmented image using BlendMask with IOS is shown in Fig. 11a and Fig. 11b, where blue polygons represented segmented wire pixels while orange rectangles were manually drawn examples of false segmented polygons. As seen, wire pixels that intertwine with apple tree branches in the back row were segmented (Fig. 12b, Fig. 12c, and Fig. 12d), which were not targets while picking apple fruit in the front row. The pixels of curly leaves and apple tree branches were usually segmented as the wire pixels falsely (Fig. 12e and Fig. 12f). A promising finding was that the segmented wire pixels were in a uniform distribution if falsely segmented wire pixels were excluded. Therefore, wire pixels needed to be further selected and processed and detail discussion was in Section 3.2.

3.2. Performance of wire reconstruction

FPEF applied in this work achieved good performance on wire reconstruction. Reconstruction results of the wires using FPEF in terms of reconstruction accuracy, missed rate and incorrect rate are shown in Table 3. FPEF achieved RA of 90.7 %, where the number of correctly reconstructed (CN) and true (N_{all}) wires was 78 and 86, respectively. The reconstruction accuracy in our study was slightly lower than that of wires on kiwifruit images (Song et al., 2021), which was 92.4 %. The MR and IR in this study were 9.3 % and 10.5 %, which were 3.2 % and 9 % higher than that in Song et al. (2021), respectively. In terms of reconstruction speed, FPEF processed 730 ms per image with a resolution of 3648 × 5472 pixels on an AMD Ryzen 7 5800X 8 core CPU, which was about slower than PPHT method (0.64 ms per image with a resolution of 512 × 341 pixels). However, our study processed images with a larger resolution and need more time. Therefore, there was only a minimal difference of performance on wire reconstruction and thus promising to identify wires in modern apple orchard using FPEF.

Additionally, it is worth discussing facts revealed by observing process of FPEF to reconstruct the wires, as shown in Fig. 13. Although IoU and PA reported by BlendMask with IOS increased, it seems to be not satisfactory because the wires are too thin and small. Segmented wire pixels were represented by blue polygons while yellow rectangles were manually drawn examples of false segmented polygons (Fig. 13a). The polygons were clustered and that with same colors belonged to same wires (Fig. 13b). The number of false wires decreased as the wires except for the bottom wires were then filtered with a constrained minimum distance. Thus, it could be argued that false reconstructed wires were more concentrated shown at the bottom of trees. However, there is no apple fruit at the bottom of the tree, which has little influence in fruit picking. From this standpoint, it is notable that the method is suitable to reconstruct the wires in modern orchard.

Reconstruction results using PPHT and polynomial fitting are shown in Fig. 14 with red and yellow lines, respectively. Both were able to obtain wire locations but polynomial fitting estimated more accurately than PPHT. Took the top wire as an example, the estimated wire location using PPHT is farther from ground truth wire than that using polynomial

Table 2
Segmentation results of BlendMask with/without IOS.

Network	IOS	IoU (%)	PA (%)	Segmentation speed (ms/image)		
				Overlap-partitioning	Model Processing	Stitching
BlendMask	No	2.01	2.12	/	2617.32	/
	Yes	43.86	61.01	0.13	80.16 × 56	54.69

Note: The model processing speed with IOS was the number of images overlap-partitioned from a raw image with a resolution of 3648 × 5472 pixels (56 in this work) multiplied by the model processing speed for overlap-partitioned images.

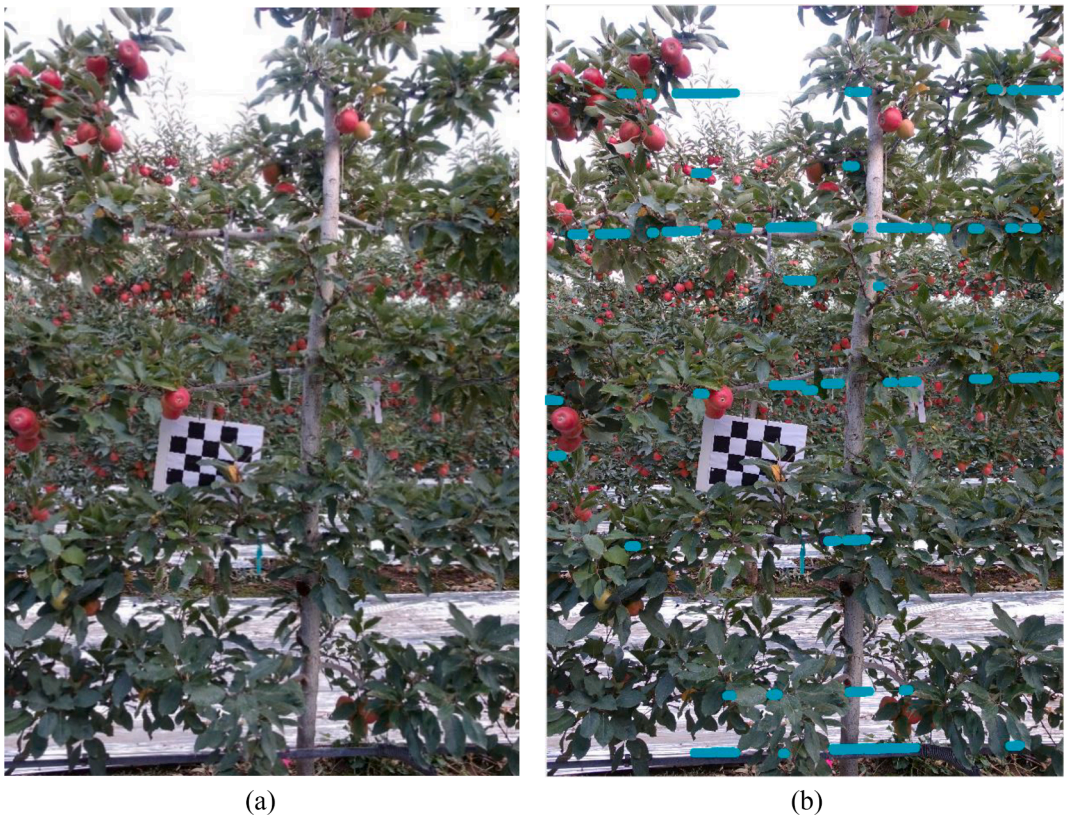


Fig. 11. An example of (a) test image; (b) segmented (stitched) image corresponding to test image shown in 'a'.



Fig. 12. Examples of segmented results using BlendMask with IOS. Blue polygons were used to depict segmented wire pixels. Orange rectangles manually drew marked examples of identified falsely wires. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Performance of wire reconstruction using FPEF.

RA (%)	MR (%)	IR (%)	Reconstruction speed (ms/image)
90.7	9.3	10.5	730

fitting. The reason was that polynomial fitting extracted more representative feature points while edge points were sampled in PPHT, which were unstandardized especially for occluded wires.

Reconstruction performance of wires was worth discussing in different occluded levels since wires are normally occluded in the natural orchard. As shown in Fig. 15a, the bottom wire was more occluded

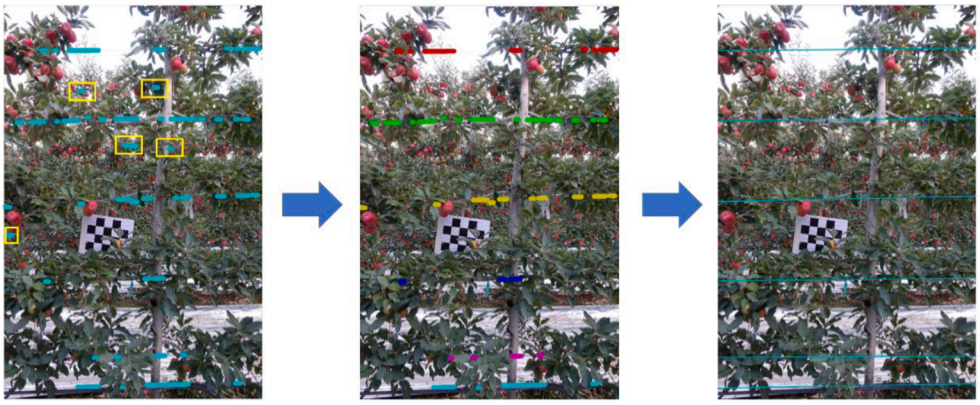


Fig. 13. An example of (a) a segmented (stitched) image; (b) an intermediate image (after cluster and filter); (c) a reconstructed image corresponding to the segmented image shown in 'a'.



Fig. 14. Reconstruction results using PPHT and polynomial fitting, respectively. (a) an image with red lines after wire reconstruction using polynomial fitting; (b) the same image with yellow lines after wire reconstruction using PPHT. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



Fig. 15. Two wires in different occluded levels (a) and their reconstruction results (b).

than the top wire, which meant less information for wire estimation. But, in fact, the methodology achieved similar reconstruction results of two wires, which laid both tend to ground truth wires (see Fig. 15b). Less information has little influence in wire estimation. Wires with different occluded level are classified as high occluded wires and low occluded wires, where wires are high occluded when the occlusion rate exceeds 50 % (visible part less than 50 %) otherwise as low occluded wires. The MRs of the high occluded wires and the high occluded wires in the testing dataset were 8.8 % and 11.1 %, respectively, which 6 out of 68 high occluded wires and 2 out of 18 low occluded wires were missed. It found that the high occluded wires had slightly lower missed rate than that of low occluded wires while the high occluded wires have less information than the low occluded wires. It showed similar advantages and practicability of the methodology for wire estimation at different levels of occlusion.

4. Conclusion

It found that the overlap-partitioned dataset with a high proportion of wire pixels makes BlendMask be sensitive to learn wire features. IOS enables to improve performance of wire segmentation, which obtained IoU and PA of 43.86 % and 61.01 % and increased by 41.85 % and 58.89

%, respectively, when measured on the full images. Wires in back row, curly leaves and apple tree branches are three main falsely segmented objects. Also, FPEF achieved a high reconstruction accuracy (90.7 %). Occluded wires are intended to be fitted using polynomial fitting rather than PPHT. Less influence on wire estimation in different occluded levels was observed in the field, considering MR values of high occluded wires slightly lower than low occluded wires. Overall, this study shows the applicability to locate wires in apple fruiting-wall architecture and thus support decision making of robots on obstacle-free picking. The purposed IOS for instance segmentation networks is novel. The methodology obtained accurate 2D location of occluded and thin wires but needs to integrated with depth information in our future work.

CRediT authorship contribution statement

Hanhui Jiang: Data curation, Investigation, Methodology, Writing – original draft. **Xiaoming Sun:** Conceptualization, Methodology, Writing – review & editing. **Wentai Fang:** Conceptualization, Methodology, Writing – review & editing. **Longsheng Fu:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Rui Li:** Methodology, Supervision, Writing – review & editing. **Fernando Auat Cheein:** Methodology, Supervision, Writing – review & editing. **Yaqoob Majeed:** Investigation, Conceptualization, 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.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (32171897); Youth Science and Technology Nova Program in Shaanxi Province of China (2021KJXX-94); Science and Technology Promotion Program of Northwest A&F University (TGZX2021-29); National Foreign Expert Project, Ministry of Science and Technology, China (DL2022172003L, QN2022172006L).

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. Biosyst. Eng. 146, 3–15. https://doi.org/10.1016/j.biosystemseng.2015.10.003.

Arikapudi, R., Vougioukas, S.G., 2021. Robotic tree-Fruit harvesting with telescoping arms: a study of linear fruit reachability under geometric constraints. IEEE Access 9, 17114–17126. https://doi.org/10.1109/ACCESS.2021.3053490.

Chen, H., Sun, K., Tian, Z., Shen, C., Huang, Y., Yan, Y., 2020. Blendmask: Top-down meets bottom-up for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp.8570–8578. https://doi.org/10.1109/CVPR42600.2020.00860.

Chen, Z., Granland, K., Newbury, R., Chen, C., 2023. HOB-CNN: Hallucination of occluded branches with a convolutional neural network for 2D fruit trees. Smart Agric. Technol. 3, 100096 https://doi.org/10.1016/j.atech.2022.100096.

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

Gené-Mola, J., Sanz-Cortiella, R., Rosell-Polo, J., Morros, J., Ruiz-Hidalgo, J., Vilaplana, V., Gregorio, E., 2020. Fruit detection and 3D location using instance segmentation neural networks and structure-from-motion photogrammetry. Comput. Electron. Agric. 169, 105165 https://doi.org/10.1016/j.compag.2019.105165.

Gené-Mola, J., Sanz-Cortiella, R., Rosell-Polo, J., Escolà, A., Gregorio, E., 2021. In-field apple size estimation using photogrammetry-derived 3D point clouds: Comparison of 4 different methods considering fruit occlusions. Comput. Electron. Agric. 188, 106343 https://doi.org/10.1016/j.compag. 2021.106343.

Hameed, K., Chai, D., Rassau, A., 2022. Score-based mask edge improvement of Mask-RCNN for segmentation of fruit and vegetables. Expert Syst. Appl. 190 (15), 116205 https://doi.org/10.1016/j.eswa.2021.116205.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp.770–778. https://doi.org/10.1109/CVPR.2016.90.

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

Hu, G., Bu, L., Chen, J., 2020. Simulation to determination of significant parameters on apple stress for combing harvesting in trellis trained trees. Sci. Hortic. 274 (15), 109654 https://doi.org/10.1016/j.scienta.2020.109654.

Jia, W., Zhang, Y., Lian, J., Zheng, Y., Zhao, D., Li, C., 2020. Apple harvesting robot under information technology: a review. Int. J. Adv. Robot. Syst. 17 (3), 1–16. https://doi.org/10.1177/1729881420925310.

Jia, W., Li, Q., Zhang, Z., Liu, G., Hou, S., Ze, J., Zheng, Y., 2021a. Optimized SOLO segmentation algorithm for the green fruits of persimmons and apples in complex environments. Trans. Chin. Soc. Agric. Eng. 37 (18), 121–127. https://doi.org/10.11975/j.issn.1002-6819.2021.18.014.

Jia, W., Zhang, Z., Shao, W., Hou, S., Ji, Z., Liu, G., 2021b. FoveaMask : a fast and accurate deep learning model for green fruit instance segmentation. Comput. Electron. Agric. 191, 106488 https://doi.org/10.1016/j.compag.2021.106488.

Long, J., Zhao, C., Lin, S., Guo, W., Wen, C., Zhao, Y., 2021. Segmentation method of the tomato fruits with different maturities under greenhouse environment based on improved Mask R-CNN. Trans. Chin. Soc. Agric. Eng. 37 (18), 100–108. https://doi.org/10.11975/j.issn.1002-6819.2021.18.012.

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., 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.

Mellor, A., Boukir, S., Haywood, A., Jones, S., 2015. Exploring issues of training data imbalance and mislabelling on random forest performance for large area land cover classification using the ensemble margin. ISPRS J. Photogramm. Remote Sens. 105, 155–168. https://doi.org/10.1016/j.isprsjprs.2015.03.014.

Niu, L., Zhou, W., Wang, D., 2017. Extracting the symmetry axes of partially occluded single apples in natural scene using convex hull theory and shape context algorithm. Multimed. Tools Appl. 76, 14075–14089. https://doi.org/10.1007/s11042-016-3781-8.

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

Song, Z., Zhou, Z., Wang, W., Gao, F., Fu, L., Li, R., Cui, Y., 2021. Canopy segmentation and wire reconstruction for kiwifruit robotic harvesting. Comput. Electron. Agric. 181, 105933 https://doi.org/10.1016/j.compag.2020.105933.

Suo, R., Gao, F., Zhou, Z., Fu, L., Song, Z., Dhupia, J., Li, R., Cui, Y., 2021. Improved multi-classes kiwifruit detection in orchard to avoid collisions during robotic picking. Comput. Electron. Agric. 182, 106052 https://doi.org/10.1016/j.compag.2021.106052.

Tang, Y., Zhou, H., Wang, H., Zhang, Y., 2023. Fruit detection and positioning technology for a Camellia oleifera C. Abel orchard based on improved YOLOv4-tiny model and binocular stereo vision. Expert Syst. Appl. 211, 118573 https://doi.org/10.1016/j.eswa.2022.118573.

UN Food & Agriculture Organization, 2022. Production of apples by the world. https://www.fao.org/faost at/en/#data. Accessed February 17, 2022.

Wan, H., Fan, Z., Yu, X., Kang, M., Wang, P., Zeng, X., 2022. A real-time branch detection and reconstruction mechanism for harvesting robot via convolutional neural network and image segmentation. Comput. Electron. Agric. 192, 106609 https://doi.org/10.1016/j.compag.2021.106609.

Wan, S., Goudos, S., 2020. Faster R-CNN for multi-class fruit detection using a robotic vision system. Comput. Netw. 168 (26), 107036 https://doi.org/10.1016/j.comnet.2019.107036.

Wang, D., He, D., 2019. Recognition of apple targets before fruits thinning by robot based on R-FCN deep convolution neural network. Trans. Chin. Soc. Agric. Eng. 35 (3), 156–163. https://doi.org/10.11975/j.issn.1002-6819.2019.03.020.

Wu, F., Duan, J., Ai, P., Chen, Z., Yang, Z., Zou, X., 2022a. Rachis detection and three-dimensional localization of cut off point for vision-based banana robot. Comput. Electron. Agric. 198, 107079 https://doi.org/10.1016/j.compag.2022.107079.

Wu, Z., Li, G., Yang, R., Fu, L., Li, R., Wang, S., 2022b. Coefficient of restitution of kiwifruit without external interference. J. Food Eng. 327, 111060 https://doi.org/10.1016/j.jfoodeng.2022.111060.

Xi, X., Xia, K., Yang, Y., Du, X., Feng, H., 2021. Evaluation of dimensionality reduction methods for individual tree crown delineation using instance segmentation network and UAV multispectral imagery in urban forest. Comput. Electron. Agric. 191, 106506 https://doi.org/10.1016/j.compag.2021.106506.

Yang, C., Wang, Z., Xiong, L., Liu, Y., Kang, X., Zhao, W., 2019. Identification and reconstruction of citrus branches under complex background based on Mask R-CNN. Trans. Chin. Soc. Agric. Mach. 50 8 22-30+69. https://doi.org/10.6041/j.issn.1000-1298.2019.08.003.

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). Comput. Electron. Agric. 155, 386–393. https://doi.org/10.1016/j.compag.2018.10.029.

Zhang, Z., Heinemann, P.H., Liu, J., Baugher, T.A., Schupp, J.R., 2016. The development of mechanical apple harvesting technology: a review. Trans. ASABE. 59 (5), 1165–1180. https://doi.org/10.13031/trans.59.11737.

Zhang, X., Karkee, M., Zhang, Q., Whiting, M.D., 2021. Computer vision-based tree trunk and branch identification and shaking points detection in Dense-Foliage canopy for automated harvesting of apples. J. F. Robot. 38 (3), 476–493. https://doi.org/10.1002/rob.21998.

Zhao, H., Qiao, Y., Wang, H., Yue, Y., 2021. Apple fruit recognition in complex orchard environment based on improved YOLOv3. Trans. Chin. Soc. Agric. Eng. 37 16 127–135. https://doi.org/0.11975/j.issn.1002-6819.2021.16.016.

Zhao, D., Lv, J., Ji, W., Zhang, Y., Chen, Y., 2011. Design and control of an apple harvesting robot. Biosyst. Eng. 110 (2), 112–122. https://doi.org/10.1016/j.biosystemseng.2011.07.005.