

HOB-CNN: Hallucination of Occluded Branches with a Convolutional Neural Network for 2D Fruit Trees

Zijue Chen, Keenan Granland, Rhys Newbury, Chao Chen*

*Laboratory of Motion Generation and Analysis, Faculty of Engineering, Monash University,
Clayton, VIC 3800, Australia*

Abstract

Orchard automation has attracted the attention of researchers recently due to the shortage of global labor force. To automate tasks in orchards such as pruning, thinning, and harvesting, a detailed understanding of the tree structure is required. However, occlusions from foliage and fruits can make it challenging to predict the position of occluded trunks and branches. This work proposes a regression-based deep learning model, Hallucination of Occluded Branch Convolutional Neural Network (HOB-CNN), for tree branch position prediction in varying occluded conditions. We formulate tree branch position prediction as a regression problem towards the horizontal locations of the branch along the vertical direction or vice versa. We present comparative experiments on Y-shaped trees with two state-of-the-art baselines, representing common approaches to the problem. Experiments show that HOB-CNN outperform the baselines at predicting branch position and shows robustness against varying levels of occlusion. We further validated HOB-CNN against two different types of 2D trees, and HOB-CNN shows generalization across different trees and robustness under different occluded conditions.

*Corresponding author

Email address: `chao.chen@monash.edu` (Chao Chen)

1. Introduction

Due to the global labor force shortage and the fast development of robots and artificial intelligence in recent years, automation in the horticulture industry has attracted attention from researchers. However, agricultural environments can be complex and unpredictable compared with industrial settings, making current agricultural tasks heavily dependent on manual labor.

The automation of agricultural tasks requires the development of robotics, artificial intelligence, and agriculture. While more robust robots and vision systems are emerging, modern farms and orchards are also training the plants to be more automation friendly. For example, the 2D fruit wall growing patterns (e.g., trellis trees¹ and Y-shaped trees, as shown in Fig. 1) has become a trend in commercial orchards in recent years. These planar tree structures simplify the automation of orchards and can benefit orchard maintenance. Therefore, current orchard automation research mainly focuses on 2D fruit wall structured trees.



Figure 1: 2D fruit wall structured trees. Y-shaped tree (left) and trellis tree (right).

Fruit orchard tasks, such as thinning, pruning, and harvesting, require an accurate tree skeleton model. However, fruit trees are often occluded by artifacts in the environment, such as foliage and fruits, which makes it a challenge for

¹Washington State University

<https://research.libraries.wsu.edu/xmlui/handle/2376/17719>

current computer vision technology (Chen et al., 2021). Furthermore, branch detection currently lacks a publicly available annotated datasets, unlike the more studied tasks of fruit detection, weed detection, and disease detection.

Branch detection is commonly done using semantic segmentation (Kang et al., 2020; Zhang et al., 2018; Majeed et al., 2020c; Chen et al., 2021; Liang et al., 2020; Granland et al., 2022). Majeed et al. (2020c) used SegNet (Badrinarayanan et al., 2017) and Liang et al. (2020) used U-Net (Ronneberger et al., 2015) to segment the visible region of both branches and trunks of a tree. Kang and Chen (2020) created a lightweight network, DasNet, to segment both apples and visible branches simultaneously. However, they are prone to produce large gaps in the output mask due to environmental occlusions. Furthermore, segmenting both visible and occluded branches has been shown to be a hard task for current neural network architectures (Chen et al., 2021). Granland et al. (2022) develop a semi-supervised approach, where an automated process would try to repair the label generated by U-Net.

To recover information from the gaps in the output mask, post-processing steps are used to hallucinate the occluded branches based on the previous detection result. Zhang et al. (2018) and Amatya et al. (2016) fit a polynomial curve to the semantic segmentation result, aiming to fill the gaps in the tree skeleton. However, this is prone to errors in false positive branch detection results. Amatya and Karkee (2016) detected the visible cherry stem and cherry clusters during the night, then used geometric relations to infer the occluded stem. Majeed et al. (2020b,a) trained a ResNet (He et al., 2015) based object detection neural networks to detect visible branches with bounding boxes. Curve fitting was applied to the center of these bounding boxes for occluded grapevine detection. This was extended in (Majeed et al., 2021) where they applied semantic segmentation to leaves that occluded the grapevines. A geometric relationship was then used to deduce the occluded section of the grapevine. However, these hallucination-based methods are completely dependent on the detection results from previous steps and post processing steps commonly do not make use of available information from the original images.

Previous work complete branch hallucination in at least two steps, in comparison, humans have achieved this autonomously, understanding that an object exists even if occluded. This is known as ‘object permanence’ and research has shown humans as young as 3-month-old (Bower, 1974; Baillargeon and DeVos, 1991) exhibit this trait. Based on the visible branches and the prior knowledge of the spatial structure, humans can deduce the coherent tree branches regardless of the occlusions. We aim to translate this intuition into a deep learning model, training it to predict both visible and occluded branches in a single step.

This work proposes a novel regression-based deep learning model, Hallucination of Occluded Branch Convolutional Neural Network (HOB-CNN), to predict the branch position of 2D fruit trees under different occlusion conditions in one step. HOB-CNN guarantees completely connected branch prediction helping to remove noise and prevent gaps in the predictions. The accuracy of HOB-CNN is compared to two current state-of-the-art (SOTA) methods. The experiments illustrated that HOB-CNN has higher accuracy in branch position prediction, and good robustness against occlusions. HOB-CNN is tested on three datasets and it shows good generalization and robustness across different tree structures and occlusion conditions.

To our best knowledge, this is the first application of an end-to-end regression-based deep learning model to predict tree position considering occlusions. The contributions of this work are threefold:

- A novel one-step regression-based deep learning model, HOB-CNN, to predict tree branch position of 2D fruit trees under occluded conditions.
- Comparative experiments between HOB-CNN and two SOTA methods in tree position prediction.
- An annotated tree position prediction dataset of Y-shaped trees in three different seasons. The dataset can be accessed [here](#).

The organization of this work is as follows: the utilised datasets are introduced in Section 2, the detailed methodology is described in Section 3, the

results and discussion are shown in Section 4 before concluding in Section 5.

2. Datasets

Three datasets were used in this work (Table 1). Firstly, a dataset containing RGB-D images of Y-shaped trees acts as the main dataset used for experiments in this work. This will be referred to as ‘Y-shaped Tree Dataset’, which includes images from different seasons, where each season will have a varying amount of environmental occlusion. We also test our approach on a dataset of trellis trees (referred to as ‘Trellis Tree Dataset’) and a dataset of grapevines (referred to as ‘Grapevine Dataset’). Both of these datasets consist of RGB images, and are used to test the generalization of HOB-CNN. The original unlabeled images of these two datasets are from AgRobotics Laboratory². Y-shaped tree and trellis tree are two prevalent 2D tree structures used in modern orchards, and grapevines are typical horizontally grown plants. These three datasets aim to provide diversity in type, structure and occlusion conditions.

Tree Type	Image Number	Image Type	Source
Y-shaped Tree	2178	RGB-D	Self
Trellis Tree	575	RGB	AgRobotics Laboratory
Grapevine	434	RGB	AgRobotics Laboratory

Table 1: Utilized tree datasets in this work.

All the images were center cropped to square and resized to 256×256 pixels for neural networks training purpose.

2.1. Y-shaped Tree

Y-shaped Tree Dataset includes 1099 winter, 518 autumn and 561 summer RGB-D images from a commercial apple orchard in northeastern Melbourne,

²Washington State University

<https://research.libraries.wsu.edu/xmlui/handle/2376/17719>

VIC. The apple species is Ruby Pink and the trees are planted east to west, with 4.125 m between the centers of each row of trees and 0.6 m between each tree in the row. A Intel RealSense D435 was used to capture RGB-D images, and the distance between the camera and the trees is between 1.8 m and 2.2 m. The trees in winter images have no occlusions from the external environment. Autumn images were collected after harvest season, when the leaves begin to fall off. Summer images were taken during harvest season, where occlusions are from both foliage and fruits. The occluded branch percentage distribution of the trees in three seasons are shown in Table 2, and an example from each season are shown in Fig. 2. We aim at predicting the position of trunk and primary branches (the branches that directly attach to the trunk) for Y-shaped trees.

Season	Image Number	Occlusion Percentage	Occlusion Amount
Winter	1099	0 ± 0	None
Autumn	518	14 ± 15	Medium
Summer	546	36 ± 9	Heavy

Table 2: Occlusion details of Y-shaped Tree Dataset in winter, autumn and summer.

The images from all seasons were randomly divided into five cross-validation dataset groups evenly, and Table 3 shows the details of each group. In each cross validation step, four groups of images were used for training and one group was used for testing. For instance, in Trial 1 the test dataset consists of Group 1 and the training dataset consists of Group 2-5.

The annotation process for Y-shaped Tree Dataset are as follows (Fig. 3):

1. The *Position Annotation* (red line) is manually labeled on the RGB images.
2. Scanning from top to bottom across the entire *Position Annotation*, for each vertical pixel row, the horizontal center locations of both the left and right branch are recorded as the position ground truth. After the branches join together in trunk segment, the only location is assigned to both left and right branches. The final HOB-CNN training target of each Y-shaped

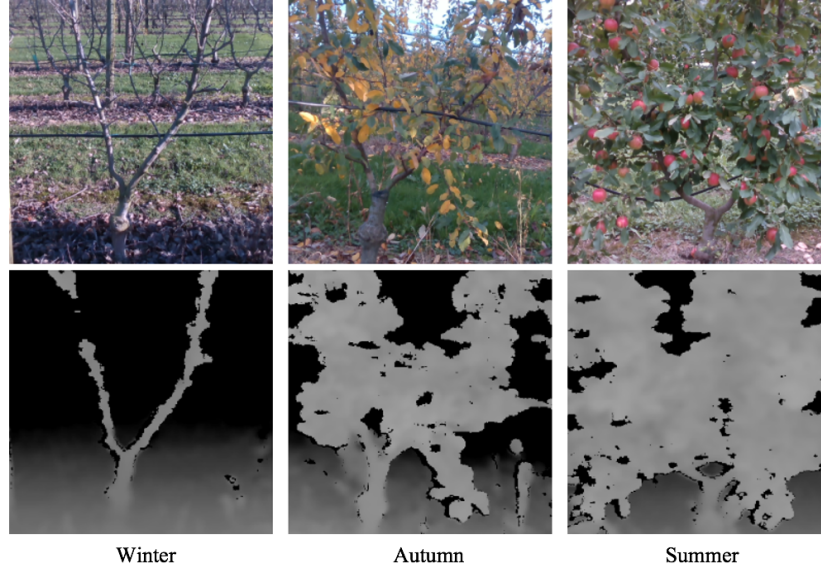


Figure 2: Example of Y-shaped trees in different seasons. RGB images on top, depth maps on bottom.

Group Number	Summer	Autumn	Winter
1	103	89	243
2	132	103	201
3	115	95	225
4	113	109	214
5	98	122	216
Total	561	518	1099

Table 3: The compositions of cross validation groups in Y-shaped Tree Dataset.

tree image is an array with the shape of $2h$, where h is the height of the image.

3. The pixel-by-pixel annotation of the trunk and primary branches was drawn on the RGB image, with the *Position Annotation* serves as reference, which is called *Whole Branch Annotation*.
4. The pixels that belonging to occlusions were removed from *Whole Branch*

Annotation, and what left is *Visible Branch Annotation*.

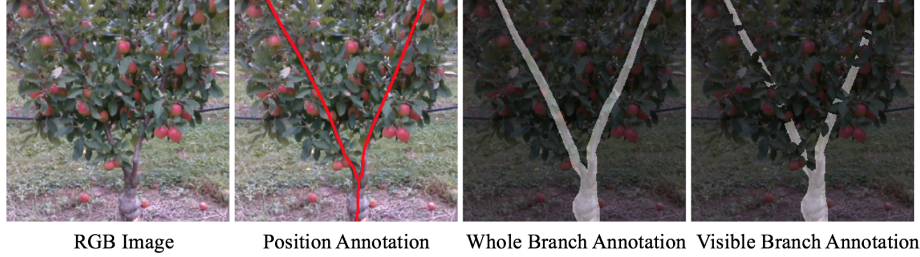


Figure 3: Example of annotations for Y-shaped Tree Dataset

2.2. Trellis Tree

Trellis tree (also called espalier tree or cordon tree) are when trees are manipulated to grow along a trellis, forming a 2D fruit wall structure. The original images of the Trellis Tree Dataset were from ‘Scifresh Apple Original and Depth-Filter RGB Images’³. This dataset includes 575 images of trellis apple trees in different levels of blurriness and lighting conditions. The 575 trellis tree images were divided into five groups for cross validation, with 115 images in each group.

For Trellis Tree Dataset (Fig. 4), the horizontal branches of the trellis trees are heavily occluded by foliage and fruits, which makes it very difficult for humans to annotate accurately. Thus, this work focuses on only tree trunk prediction. The generation of HOB-CNN training targets follows the same process as Y-shaped Tree corresponding to the *Position Annotation*.

2.3. Grapevine

Grapevine Dataset includes images of grapevines from early spring and summer (Fig. 5). The original 184 grapevine images in early spring are from ‘Vineyard Canopy Images during Early Growth’ Stage⁴, and 250 original summer grapevine images are from ‘Full Stages of Wine Grape Canopy and Clusters’⁵.

³<http://hdl.handle.net/2376/17720>

⁴<http://hdl.handle.net/2376/16939>

⁵<http://hdl.handle.net/2376/17628>

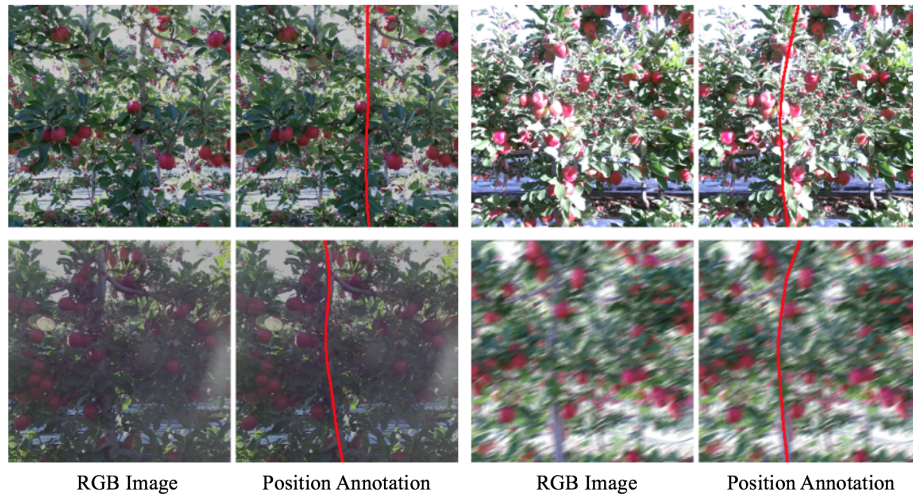


Figure 4: Annotation examples for Trellis Tree Dataset

The grapevine images from early spring have minimal occlusions from the early stage shoots. Summer grapevines are heavily occluded by leaves. Additionally, the summer dataset contains images with different camera angles, lighting conditions and maturity, which increased the diversity of summer images.

To make up the limited training images of the Grapevine Dataset and generate more training images, an augmentation process was applied to each image, square cropping from top, center and bottom with a side length equal to $0.8 \times H$, where H is the height of the original images.

Then, all the cropped images were distributed into five cross validation groups, the details of each group are shown in Table 4.

3. Methodology

3.1. HOB-CNN

HOB-CNN was designed to predict the horizontal location of the branch along a vertical direction, or vice versa. This is applicable to both vertical trees (e.g. Y-shaped trees, trellis trees) and horizontal vines (e.g. grapevines). However, if the tree has multiple branches, there will be multiple locations.

Group Number	Early Sprint	Summer
1	114	147
2	99	162
3	114	147
4	113	147
5	112	149
Total	552	752

Table 4: The compositions of cross validation groups in Grapevine Dataset

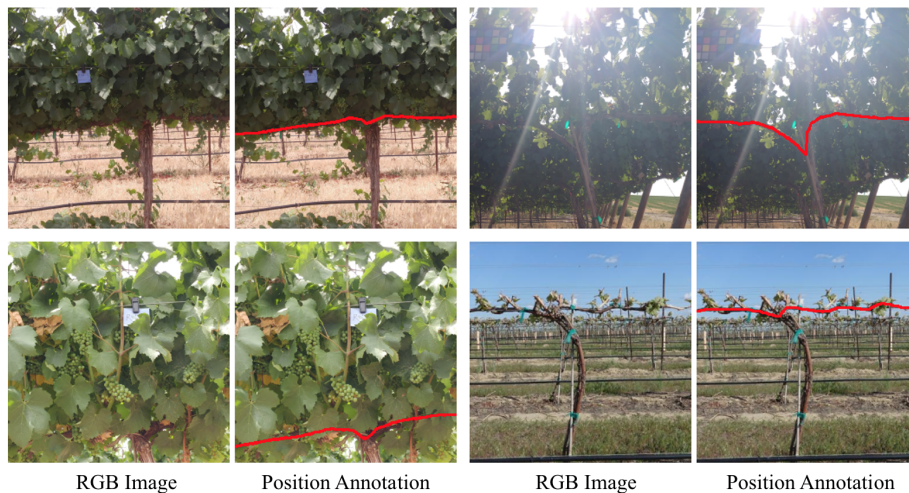


Figure 5: Annotation examples for Grapevine Dataset

Therefore, HOB-CNN will output one horizontal location for each branch at each pixel row. For example, the prediction of Y-shaped trees has two horizontal locations in each vertical pixel row, one for the left branch and one for the right branch.

HOB-CNN consists of a backbone, followed by three fully connected layers. We used EfficientNet-b4 (Tan and Le, 2019) as the backbone, which has SOTA performance on image recognition with a smaller amount of parameters compared to similarly performing networks. The output of the backbone was flattened, then fed into dense layers (see Fig. 6). The first and second dense

layers have 2048 and 512 neurons, respectively, with ReLU activation. Finally, the last dense layer outputs $n \times h$ neurons, where n is the number of branches in the tree, and h is the height/width of the image in pixels.

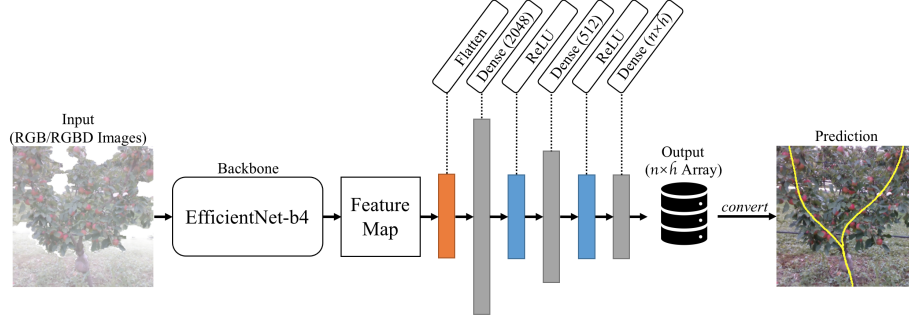


Figure 6: HOB-CNN architecture

3.2. Baselines: U-Net with Curve Fitting

U-Net (Ronneberger et al., 2015) is a fully convolutional neural network for semantic segmentation. In our previous work (Chen et al., 2021), it was found that U-Net outperformed other SOTA methods at predicting occluded tree branches. Therefore we used the same U-Net model, with ResNet-34 (He et al., 2015) as the encoder. U-Net was used to train two different models.

1. **U-Net Visible** to detect only *visible* branch sections
2. **U-Net Whole** to hallucinate the *whole* branch under occlusions.

Curve Fitting is a common post-processing method in branch detection (e.g (Zhang et al., 2018; Majeed et al., 2020b,a, 2021)), and was utilized in this work to create a continuous curve from a possibly noisy semantic mask from *U-Net Visible* or *U-Net Whole*. In this work, ‘*U-Net Visible* + curve fitting’ method is referred to as *Visible CF*, and ‘*U-Net Whole* + curve fitting’ is referred to as *Whole CF*.

The semantic predictions from the two U-Net models were converted to a single-pixel width skeleton in a multi-step process. Firstly, filters were used to reduce noise, followed by fitting a curve to the result. Two curve fitting

methods were tested: cubic splines and n th polynomial (3rd to 15th). The 5th order polynomial curve fitting was used as it had the best performance. The process is as follows (Fig. 7):

- A blob filter was applied removing blobs with an area less than 65 pixels.
- Waypoints were calculated for curve fitting by converting the binary label to a series of points, taking the center blob position of each row.
- The waypoints were then split into a 'Left' and 'Right' path.
- After the 'Left' and 'Right' path were generated, a 5th order polynomial curve was fitted to each path and used to generate the output label.

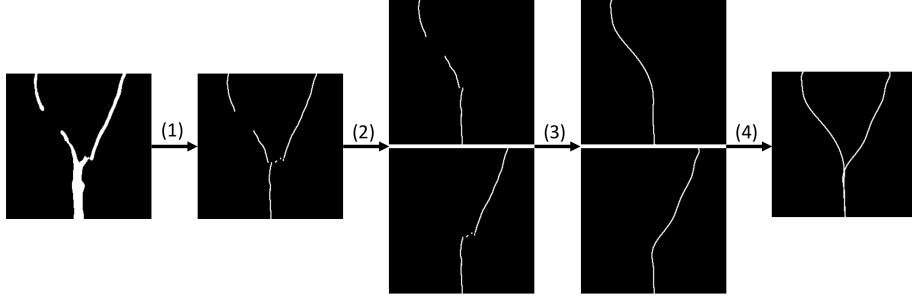


Figure 7: U-Net curve fitting procedure: (1) remove noise and extract waypoints; (2) split waypoints to 'Left' and 'Right'; (3) fit curve on waypoints; (4) combine fitted curves together.

3.3. Training Details

U-Net Visible and *U-Net Whole* were trained on the three datasets separately (RGBD input images for Y-shaped Tree Dataset and RGB input images for Terllis Tree and Grapevine Datasets), with a batch size of 8 for 50 epochs, and a horizontal flip was used in augmentation. Weighted dice loss (Sudre et al., 2017) and the Adam optimizer were utilized.

Similarly, HOB-CNN was trained by each dataset separately with a batch size of 8, and horizontal flip was used in augmentation. Mean Squared Error (L2 loss) was used as the loss function and the Adam optimizer was utilized.

The model was evaluated after convergence and before overfitting was observed, which is 90 epochs for Y-shaped Trees Dataset, 400 epochs for Trellis Trees Dataset and 300 epochs for Grapevine Dataset.

3.4. Metrics

Root Mean Squared Error (RMSE) and Correlation Coefficient (r) were adopted for evaluating the accuracy of the predicted branch positions of *Visible CF*, *Whole CF* and HOB-CNN.

Root Mean Squared Error (RMSE): RMSE was used in this work to measure the error between the tree position prediction and ground truth:

$$RMSE = \sqrt{\frac{\sum_{n=1}^N (g_i - p_i)^2}{N}} \quad (1)$$

where N is the number of data points, g_i and p_i are the ground truth value and the predicted value respectively.

Correlation Coefficient (r): Correlation coefficient is a widely used metric for regression analysis. It is defined as the degree of relationship between two groups of data. Correlation coefficient is evaluated between $[-1, 1]$, where 1 means the two variables have perfect correlation and -1 means the two variables are in perfect opposites, 0 indicates they are not related. The equation is as follows:

$$r = \frac{\sum (p_i - \bar{p})(g_i - \bar{g})}{\sqrt{\sum (p_i - \bar{p})^2 \sum (g_i - \bar{g})^2}} \quad (2)$$

where p_i and $\bar{p}$ are the predicted values and the mean of the predicted values; g_i and $\bar{g}$ are the ground truth values and the mean of the ground truth values.

4. Results and Discussions

In Section 4.1, we first discussed the performance of HOB-CNN at detecting the branch positions of Y-shaped trees and compared its results with *Visible CF* and *Whole CF*. We analyzed the impact of occlusion on their performance in Section 4.2. In Section 4.3, processing time and annotation time of each

method are compared. HOB-CNN is also applied on branch detection of trellis trees and grapevines in Section 4.4. Finally, we analyzed the common features of HOB-CNN predictions with large errors in Section 4.5.

4.1. Y-shaped Tree Position Prediction

In the experiment of Y-shaped tree prediction, HOB-CNN shows good performance at hallucinating complete branch paths under different occlusions. Some examples are shown in Fig. 8, where the yellow curves represent branch center lines that HOB-CNN predicted while the underling red curves are the ground truth.



Figure 8: HOB-CNN prediction examples on Y-shaped tree at summer (left), autumn (middle) and winter (right). Red lines represent the GT.

To quantify the accuracy of HOB-CNN, RMSE and correlation coefficient are utilized for evaluation. The results in different seasons are compared with the two baselines, *Visible CF* and *Whole CF*. As shown in Table 5, HOB-CNN has the highest accuracy in all seasons. This advantage is more evident in summer when the occlusion is heavy: the mean RMSE of HOB-CNN in summer is 2.671 pixels, obviously lower than the widely used *Visible CF* (5.651 pixels) and *Whole CF* (4.151 pixels). Comparing *Visible CF* and HOB-CNN, the standard deviations of RMSE and correlation coefficient drops from 6.235 to 1.589 and from 0.092 to 0.032 respectively. This shows that HOB-CNN’s performance is more stable under different occlusion scenarios.

Figure 9 shows some examples of *Visible CF*, *Whole CF* and HOB-CNN at tree position prediction. The semantic predictions of visible branch and

Metric	Season	Visible CF	Whole CF	HOB-CNN
RMSE	Winter	2.622 $\pm$ 4.020	2.719 $\pm$ 3.741	1.962$\pm$1.290
	Autumn	3.312 $\pm$ 4.546	3.421 $\pm$ 6.839	2.463$\pm$2.023
	Summer	5.651 $\pm$ 9.705	4.151 $\pm$ 5.755	2.671$\pm$1.549
	Total	3.566 $\pm$ 6.235	3.255 $\pm$ 5.201	2.264$\pm$1.589
Correlation Coefficient	Winter	0.992 $\pm$ 0.055	0.992 $\pm$ 0.068	0.996$\pm$0.020
	Autumn	0.979 $\pm$ 0.093	0.979 $\pm$ 0.097	0.989$\pm$0.053
	Summer	0.969 $\pm$ 0.138	0.984 $\pm$ 0.073	0.995$\pm$0.024
	Total	0.983 $\pm$ 0.092	0.987 $\pm$ 0.077	0.994$\pm$0.032

Table 5: RMSE and correlation coefficient of *Visible CF*, *Whole CF* and HOB-CNN in Y-shaped tree position prediction.

whole branch are also shown in the images. The semantic predictions of *U-Net Whole* shows its potential of hallucinating occluded branches across various occlusion conditions, but can not guarantee the fully connectivity of the tree structure. However, *U-Net Whole* tries to deduce the occluded branch segments with the visible branches as reference, and the accurate hallucinations offer more reference information for curve fitting, which can potentially explain the improved accuracy from *Visible CF* to *Whole CF* during summer in Table 5.

It can also be seen from Fig. 9 that the semantic predictions of both *U-Net Visible* and *U-Net Whole* in summer and autumn are more prone to noises and disconnections because of occlusions. These noises and disconnections introduce errors and lead to deviations in curve fitting results. On the other hand, low order curves cannot represent sharp bends accurately, whereas high order curves are prone to overfitting and are likely to generate too many bends. Even though in our experiments, a 5th order polynomial curve shows the best fitting results, the details of the predicted tree branches are also limited by the curves. For example, the sharp bend that happens at the merging point as shown in Fig. 9 (top), cannot be accurately represented using a 5th order polynomial. In contrast, HOB-CNN infers branch positions directly on the original image,

without using curve fitting. Therefore, HOB-CNN has more complete reference information, which increases the prediction accuracy. Also, as HOB-CNN is not restricted by curve fitting, it is more flexible at predicting branch structure, and can better fit the branch path where sharp bend occurs.



Figure 9: Examples of *Visible CF*, *Whole CF* and HOB-CNN at Y-shaped tree position prediction. The red curves represent the ground truth annotation.

4.2. Relationship between Performance and Branch Occlusion

To better demonstrate the influence of occlusion on tree position prediction, we show the trend of RMSE and correlation coefficient of *Visible CF*, *Whole CF* and HOB-CNN with increasing occlusion in Fig. 10. *Visible CF* is the most vulnerable to occlusion. It outperforms *Whole CF* under no occlusions, but the prediction accuracy decreases rapidly as the occlusion increases, and is surpassed by *Whole CF* at 5% occlusion. HOB-CNN has the most robust performance, consistently outperforming the two baselines in both RMSE and correlation coefficient with the smallest error and variance throughout.

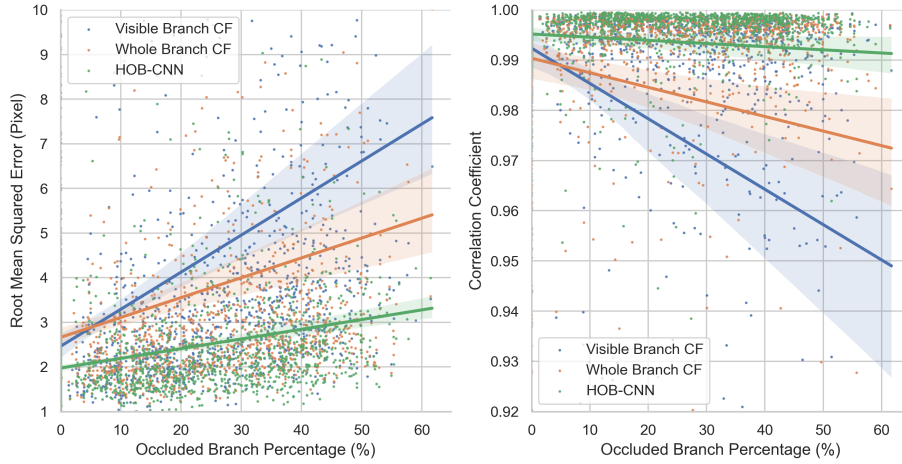


Figure 10: RMSE and correlation coefficient on varying Occluded Branch Percentage. The lines show the fitted trend of the data while the highlighted area represents the variance.

4.3. Processing and Annotation Time Comparison

Table 6 summarises the average processing time and annotation time for all three methods. *Visible CF* and *Whole CF* require post processing steps like filtering noises, extracting way points, as well as fitting curves. These extra steps add to the overall processing time. The neural network predicting times of both *Visible CF* and *Whole CF* are very close to that of HOB-CNN’s, while most of the processing time is spent on curve fitting. The total processing times of *Visible CF* and *Whole CF* are 1451ms and 1716ms respectively. This is because

Whole CF needs to extract more way points for curve fitting than *Visible CF*, so that the curve fitting time increases by 19%. HOB-CNN, in comparison, only takes 58ms, about 3% of the two baselines, giving it advantage for fast detection in field.

Model	Visible CF	Whole CF	HOB-CNN
Processing Time (ms)			
Neural Network	53	52	58
Curve Fitting	1397	1663	-
Total	1451	1716	58
Annotation Time (s)			
Winter	90	90	15
Autumn	160	120	15
Summer	210	150	20
Average	153	120	17

Table 6: Processing and annotation times for *Visible CF*, *Whole CF* and HOB-CNN.

The required effort of ground truth labeling is another important consideration for model choosing. *Visible CF* and *Whole CF* require the semantic annotation, whereas HOB-CNN only requires the center line to represent the branch positions, so HOB-CNN saves labeling time greatly. Besides, since *Visible CF* requires identifying non-occluded branch segments and accurately labeling them, it takes extra time than *Whole CF*. From the table, The annotation time of HOB-CNN is 17s per image averagely, compared with *Visible CF*'s 153s and *Whole CF*'s 120s, it saves more than 85% annotation time. Especially for summer trees, changing the model from *Visible CF* to HOB-CNN makes the annotation time drop from 210s to 20s per image.

4.4. HOB-CNN on Other Datasets

We also apply HOB-CNN on predicting the trellis tree trunks and grapevine branches. By changing the number of neurons in the output layer, HOB-CNN

can be applied to different types and structures of 2D trees. For example, the number of neurons in the output layer of the trellis trees model reduces neuron from 512 to 256 compared to the Y-shaped tree model.

Trial	Metrics	
	RMSE	Correlation Coefficient
1	1.398±0.955	0.951±0.081
2	1.831±0.845	0.938±0.100
3	1.530±0.842	0.960±0.062
4	1.463±0.786	0.948±0.122
5	1.927±0.945	0.945±0.073
Average	1.630±0.902	0.948±0.091

Table 7: Performance of HOB-CNN on trellis tree trunk detection

Trellis Tree Dataset: Table 7 illustrates that HOB-CNN has a consistent performance in detection of trellis tree trunks. Some prediction examples are shown in Fig. 11. The results show the robustness of HOB-CNN under different lighting and blurriness conditions.

Trial	Early Spring		Summer	
	RMSE	Correlation Coefficient	RMSE	Correlation Coefficient
1	1.759±0.606	0.946±0.035	2.123±1.158	0.964±0.028
2	2.794±1.077	0.862±0.140	3.282±1.793	0.895±0.152
3	1.869±0.880	0.950±0.040	2.461±1.766	0.948±0.113
4	2.706±3.892	0.921±0.057	2.706±0.966	0.933±0.062
5	2.337±1.439	0.935±0.052	2.672±1.618	0.947±0.063
Average	2.279±2.032	0.925±0.079	2.661±1.554	0.937±0.099

Table 8: Performance of HOB-CNN on grapevine branch detection

Grapevine Dataset: The performances of HOB-CNN at grapevine detec-

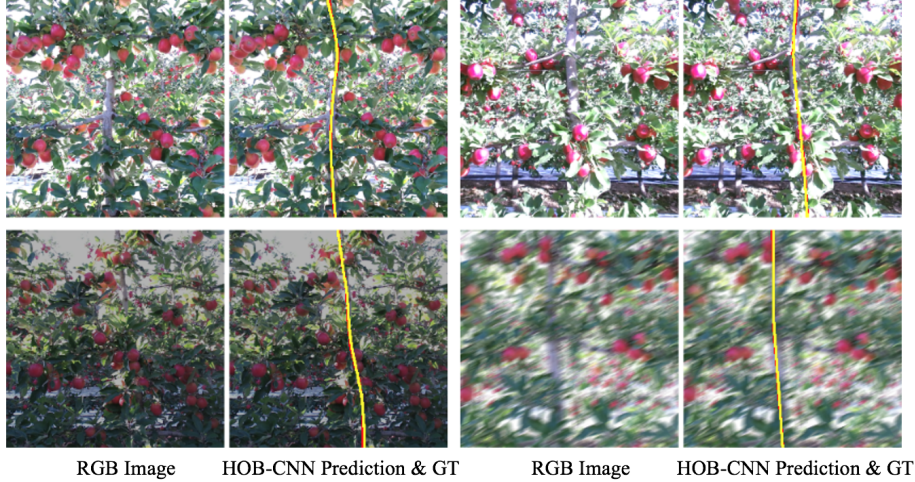


Figure 11: HOB-CNN prediction examples on Trellis Tree Dataset. The left images are RGB image and the right images are the prediction (yellow curves) overlay with ground truth (red curves).

tion are shown in Table 8. From early spring to summer, the RMSE increases from 2.279 to 2.661 as a result of increasing occlusions. Contrary to intuition, the correlation coefficient in summer is 0.937, which is slightly higher than that in early spring (0.925). This can potentially be explained by the larger amount and diversity of the summer dataset, which results in the improved performance on summer images. Some examples are shown in Fig. 12.

Trellis Tree and Grapevine Datasets are only utilized to test HOB-CNN on other 2D-structured fruit trees. Both the number and the diversity of these two datasets are limited. However, the experiments show HOB-CNN’s ability at detecting different 2D trees. In the future, more types of trees and more occlusion conditions will be considered.

4.5. HOB-CNN Error Analysis

We analyze HOB-CNN predictions which have a RMSE larger than 4 pixels from all three datasets. Common features of these large error images and their corresponding image numbers are summarized in Table 9 and the details are described below. The examples are shown in Fig. 13.

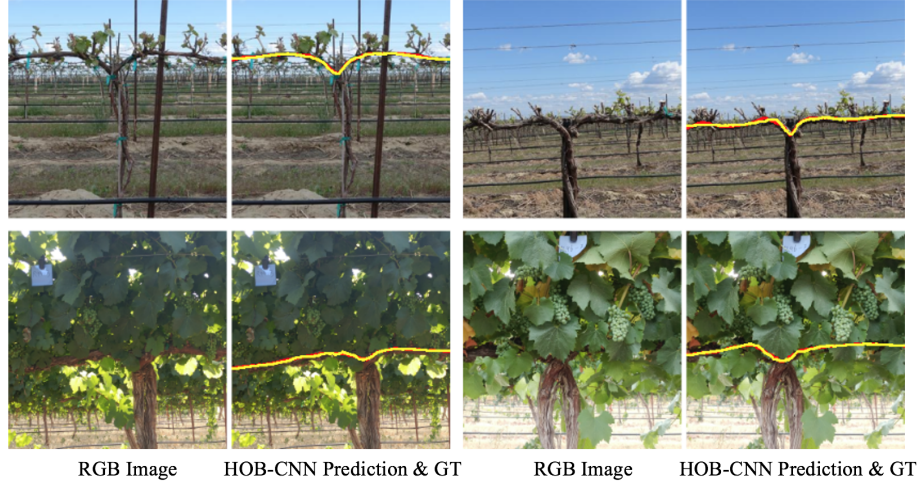


Figure 12: HOB-CNN prediction examples on Grapevine Dataset. Yellow curves are the predictions and red curves are ground truth.

Common Feature	Number of image		
	Y-shaped Tree	Trellis Tree	Grapevine
Extremely Occluded Tree	47	2	43
Thin Branch	7	0	0
Blurry Image	2	9	0
Unusual Tree Shape	16	2	0
Neighbouring/Background Tree	8	0	0
Overexposed Image	0	0	3
Sharp Bend Tree	27	0	38
Others	40	2	22
Total	148	14	106
Dataset Percentage	6.80%	2.40%	24.40%

Table 9: Common features of the large error images and their corresponding image numbers in Y-shaped Tree, Trellis Tree and Grapevine Datasets.

Extremely Occluded Tree situations involves continues large parts of the branch being occluded. In these situations, the branch cannot be easily found

due to the occlusion by the fruits and leaves in the picture. This problem might be resolved with more training data, or by matching trees against less occluded seasons. Both of these will be left to future work.

Thin Branch only appears in Y-shaped trees. In these images, the thin branches often only has one or two pixels width, which makes it hard for the neural network to extract the features.

Blurry Picture has less detail information, resulting in inferior predictions. In practice, blurry pictures can be avoid easily by stabilizing the camera.

Unique Tree Shape (e.g. long trunk, close branches, sloping tree) have less images in training datasets than normal shaped trees, which contributes the bias of the vision system towards the normal shaped trees. This can be fixed by including more unique shaped trees into training dataset.

Neighbouring/Background Tree situations involve disturbing branches from neighbouring or background trees. The same features of the branches tricked the vision system to mistake the wrong branch.

Overexposed Image is similar to blurry image. Details are washed out by the highlights, and thus there are less details in the image. This can be fixed by adding polarized filter lens to the camera under strong lights.

Sharp Bend Tree happens in Y-shaped trees and grapevines. The prediction around the sharp bend are less accurate than around smooth branch. This can potentially be explained by the bias towards smooth branches in the training material, in which smooth branches have much more images than sharp bend branches.

5. Conclusion

This work proposes a novel single-step regression-based deep learning model, HOB-CNN, for predicting branch position of 2D structured fruit trees under different occlusion conditions. This work shows the potential of regression models in foliage tree branch prediction. The comparative experimental results indicate it outperforms two baselines that represent the common approaches for occluded

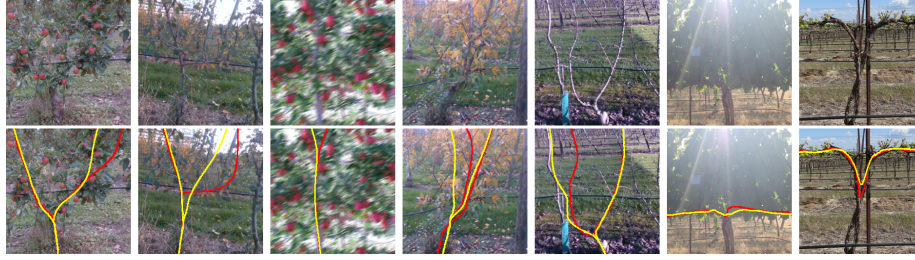


Figure 13: Examples of large error predictions. Top row: RGB images; bottom row: predictions (yellow curves) overlay with ground truths (red curves). From left to right: Extremely Occluded Tree, Thin Branch, Blurry Image, Unusual Tree Shape, Neighbouring/Background Trees, Overexposed Image and Sharp Bend Tree.

branch hallucination with higher accuracy and better robustness to occlusions. Furthermore, HOB-CNN shows the ability to generalize across different trees.

However, during harvest season, the occlusions on the branches can be so heavy that occasionally even humans are unable to recognize the branches on the images. Thus, it is very challenging to manually annotate the training targets, leading to the imbalanced Y-shaped Tree Dataset (more winter tree images) in this work. In future work, more data is going to be collected to make the dataset balance. Furthermore, we will train the vision system to predict the heavily occluded branches by providing an additional input of lightly occluded images of the same tree. Additionally, we will assign the thickness profile to each branch position prediction, aiming to accurately reconstruct 3D models of the tree.

6. Acknowledgments

We thank for the support by ARC Nanocomm HUB IH150100006 and Fankhauser Apples.

References

Amatya, S., Karkee, M., 2016. Integration of visible branch sections and cherry clusters for detecting cherry tree branches in dense foliage canopies. Biosys-

- tems Engineering 149, 72–81.
- Amatya, S., Karkee, M., Gongal, A., Zhang, Q., Whiting, M. D., 2016. Detection of cherry tree branches with full foliage in planar architecture for automated sweet-cherry harvesting. *Biosystems Engineering* 146, 3–15, special Issue: Advances in Robotic Agriculture for Crops.
- Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (12), 2481–2495.
- Baillargeon, R., DeVos, J., 1991. Object permanence in young infants: Further evidence. *Child development* 62 (6), 1227–1246.
- Bower, T. G., 1974. Development in infancy. WH Freeman.
- Chen, Z., Ting, D., Newbury, R., Chen, C., 2021. Semantic segmentation for partially occluded apple trees based on deep learning. *Computers and Electronics in Agriculture* 181, 105952.
- Granland, K., Newbury, R., Chen, Z., Ting, D., Chen, C., 2022. Detecting occluded y-shaped fruit tree segments using automated iterative training with minimal labeling effort. *Computers and Electronics in Agriculture* 194, 106747.
- URL <https://www.sciencedirect.com/science/article/pii/S0168169922000643>
- He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. *CoRR abs/1512.03385*.
- Kang, H., Chen, C., 2020. Fruit detection, segmentation and 3d visualisation of environments in apple orchards. *Computers and Electronics in Agriculture* 171, 105302.
- Kang, H., Zhou, H., Chen, C., 2020. Visual perception and modeling for autonomous apple harvesting. *IEEE Access* 8, 62151–62163.

- 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. *Computers and Electronics in Agriculture* 169, 105192.
- Majeed, Y., Karkee, M., Zhang, Q., 2020a. Estimating the trajectories of vine cordons in full foliage canopies for automated green shoot thinning in vineyards. *Computers and Electronics in Agriculture* 176, 105671.
- Majeed, Y., Karkee, M., Zhang, Q., Fu, L., Whiting, M. D., 2020b. Determining grapevine cordon shape for automated green shoot thinning using semantic segmentation-based deep learning networks. *Computers and Electronics in Agriculture* 171, 105308.
- Majeed, Y., Karkee, M., Zhang, Q., Fu, L., Whiting, M. D., 2021. Development and performance evaluation of a machine vision system and an integrated prototype for automated green shoot thinning in vineyards. *Journal of Field Robotics* 38 (6), 898–916.
URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.22013>
- Majeed, Y., Zhang, J., Zhang, X., Fu, L., Karkee, M., Zhang, Q., Whiting, M. D., 2020c. Deep learning based segmentation for automated training of apple trees on trellis wires. *Computers and Electronics in Agriculture* 170, 105277.
- Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. *CoRR* abs/1505.04597.
- Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., Jorge Cardoso, M., 2017. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *Lecture Notes in Computer Science*, 240–248.
- Tan, M., Le, Q., 09–15 Jun 2019. EfficientNet: Rethinking model scaling for convolutional neural networks. In: Chaudhuri, K., Salakhutdinov, R. (Eds.), *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97

of Proceedings of Machine Learning Research. PMLR, pp. 6105–6114.

URL <https://proceedings.mlr.press/v97/tan19a.html>

Zhang, J., He, L., Karkee, M., Zhang, Q., Zhang, X., Gao, Z., 2018. Branch detection for apple trees trained in fruiting wall architecture using depth features and regions-convolutional neural network (r-cnn). Computers and Electronics in Agriculture 155, 386 – 393.