

Side-view apple flower mapping using edge-based fully convolutional networks for variable rate chemical thinning

Xu (Annie) Wang, Julie Tang, Mark Whitty*

School of Mechanical and Manufacturing Engineering, UNSW Sydney, Australia



ARTICLE INFO

Keywords:

Flower detection
Flower density mapping
Semantic segmentation
FCN
Apples

ABSTRACT

Apple trees commonly require the removal of excessive flowers by thinning to produce high quality fruit. Machine vision has recently been applied to detect the flower density as the first step in this process. Existing work relying on color thresholding is sensitive to imaging conditions and the most recent published work using deep learning in this context has proven to be exceptionally slow to process. This paper presents an apple flower segmentation method on a pixel level based on a Fully Convolutional Network (FCN) together with a process of generating a map that can be used for a variable rate chemical sprayer. Despite the challenging conditions of an uncontrolled environment, our apple flower detector was able to generate a F_1 score at pixel-level up to 85.6%, which is a relatively high accuracy in terms of pixel-level segmentation. Our method has been tested on both daytime and night-time datasets, which strongly validates the ability of our apple flower detector to work under different conditions. The resulting detections are georeferenced and merged into a density map in the format necessary for application by a variable rate chemical sprayer. Finally, this flower density mapping system will benefit farmers by visualising the whole crop and extracting useful information to support their decision making for chemical thinning.

1. Introduction

Development of sensing and localization technology in robotics has made farm management more precise and efficient, enabling management of on-farm variability. Therefore, advanced computer vision combined with localization devices has been applied in orchard management, such as yield estimation, flower counting and fruit harvesting. Such technology has the capability to replace traditional and trivial human inspection needed on farms, providing more accurate results as well as significantly reducing human labor requirements and therefore reducing costs. In addition, creating these kinds of systems also make it possible to provide decision support down to tree-level or even area-level. Such management is essential as growth through the orchard is variable between trees.

Apples rank third in global fruit production with about 86 million metric tons produced in 2018, and rank fourth behind potatoes, oranges and tomatoes in the Australian horticulture industry. In 2016–17, Australia produced close to 320,000 tonnes of apples, worth approximately \$AUD 497 million (Australian Government Department of Agriculture and Water Resources, 2018a). Although apple crops are high value, there are still some problems, such as poor fruit size, biennial bearing and seasonal labor availability (Australia) (Australian

Government Department of Agriculture and Water Resources, 2018b).

To produce marketable fruit, the crop load must be controlled through flower and fruit thinning (Aggelopoulou et al., 2010). Chemical thinning during blooming is the first step towards removing excess crop load and at the same time helps break the biennial habit (Williams, 1979). When chemical thinning is inadequate, labor intensive hand-thinning is required (Aggelopoulou et al., 2010). Given hand-thinning processes are exceptionally labor-intensive and the second highest labor cost behind harvesting (nine times higher than chemical thinning), improved chemical thinning processes are desired (Bound, 2018).

Apple flower density has been closely correlated to yield (Aggelopoulou et al., 2010). Thus, apple flower density is considered to be an important index to guide thinning procedures. The most common practice currently in Australian apple industry is that farmers manually count the number of flowers in some randomly sampled trees within the orchard and use their experience to decide what level of thinning is needed for the whole orchard, which is very time-consuming, labor-intensive, and inaccurate. As individual tree loads are known to be variable, application of chemical thinners at a fixed rate for an entire orchard block can lead to significantly over-thinned or under-thinned trees. Since only a limited sample of trees are inspected, large errors in estimation are possible, especially when it comes to individual trees.

* Corresponding author.

E-mail address: m.whitty@unsw.edu.au (M. Whitty).

Thus, capturing variation within the orchard block is integral to improved chemical thinning practices and reduced hand-thinning requirements. This therefore created a sense of urgency to develop an end-to-end autonomous flower density estimation system using computer vision techniques, which will produce more accurate results as well as reduce labor requirements to tackle all problems above.

Despite its importance, there has been relatively limited research in apple flower density estimation. Most of the research work relies on simple color thresholding under specific lighting conditions (Hočvar et al., 2014; Thorp and Dierig, 2011; Dorj et al., 2012), or placing a backing board to remove the background (Aggelopoulou et al., 2011), which are not applicable or robust in large commercial apple orchards. One more recent method deploying deep learning has greatly improved accuracy over thresholding methods, however, the method takes 50 seconds (Dias et al., 2018b) to analyse a single tree which limits its applicability to mapping of whole orchards within a reasonable time-frame.

In this paper, we propose an autonomous apple flower mapping system for a variable rate chemical sprayer using deep learning. Our main contributions are listed in the following:

- A novel end-to-end flower segmentation algorithm improved from the FCN8s network, which delivers a relatively high F_1 score for small and complex objects with relatively high speed.
- A thorough evaluation of the algorithm on both night and day time datasets captured on a moving vehicle in a commercial apple orchard with central leader trees.
- A novel pipeline for autonomous apple flower density mapping using results from our flower segmentation algorithm for the purpose of mapping density variation in an orchard for improved chemical thinning practices.

The remainder of the paper is organized as follows. Section 2 discusses the most relevant existing approaches for flower and fruit detection. Section 3 describes our camera system and data collection as well as our proposed apple flower segmentation algorithm and density mapping process. Related training procedures and evaluation metrics are also described in this section. Results on different segmentation methods as well as different datasets are shown in Section 4. Discussion and some future work are described in Section 5. Our concluding remarks are presented in Section 6.

2. Related work

Although flower (or bloom) density estimation is important, there has been relatively limited research in this field. The most common and traditional method for flower segmentation is by color thresholding (Aggelopoulou et al., 2011; Hočvar et al., 2014; Thorp and Dierig, 2011; Dorj et al., 2012) in different color spaces such as HSV (Hue, Saturation, Value) and RGB. This method is limited to controlled conditions, such as controlled illumination or placing a backing board to remove background noise from adjacent rows and the sky.

Due to the success of CNN in image classification, Dias et al. (2018a) introduced a CNN-based algorithm for apple flower detection to classify apple flower superpixels. However, this method is highly reliant on the accuracy of superpixel segmentation for flower clusters, with a pixel-level F_1 score reported of only 67.8% in their later work (Dias et al., 2018b). Dias et al. (2018b) then used DeepLabV2_Resnet101 followed by a post-processing Region Growing Refinement (RGR), which was able to achieve a high F_1 score of 83.3% on hand-held images at high resolution of 18 megapixels, however, when evaluating on images with smaller number of pixels per flower captured with a backing board, the F_1 score was reduced to 77.3%. Although the flower detection was satisfactory, the 50 second time reported for estimating one full tree is impractical for density mapping of the entire orchard since one orchard

can contain thousands of trees. Farjon et al. (2019) used Faster-RCNN to count apple flower clusters to estimate the exact number for individual trees, however, their method was only evaluated on 5 images containing 819 flowers with detection based on hand-held 20 mega-pixel images.

In other horticultural crops, color thresholding with controlled conditions is still a common method for almond flower segmentation (Underwood et al., 2016) and peach flower segmentation (Horton et al., 2017). Detection accuracy was not mentioned in the first work but it can be seen that some overexposed branches were falsely detected, while the second work mentioned an average detection rate of 84.3% that was not a pixel-level evaluation. There are also traditional machine learning methods used to classify flowers on preliminary results generated from color thresholding. Recently, Wang et al. (2018) proposed a machine learning protocol consisting of Speeded Up Robust Features (SURF) and Support Vector Machine (SVM) classification for mango flower segmentation, but the F_1 score at pixel level was only about 56%, which might be due to using a small-size dataset and shows traditional methods are insufficiently robust.

Besides research on horticulture, research in other fields on semantic segmentation is also useful. Ma et al. (2020) proposed a two-stage wheat ear segmentation method base on FCN8s. To overcome the difficulties of classifying wheat ear from similar looking canopy, input images first went through a coarse segmentation utilizing superpixels, similar to Dias et al. (2018a), and then were refined by FCN8s later, which gave a relatively high F_1 of 83.7%. Wu et al. (2018) also proposed a multitask framework based on FCNs to separately segmenting roofs of buildings and predicting outlines of roofs from aerial images, which also shows a high F_1 score of 86.9%, which gives us the inspiration of utilizing the edge class in our work.

The system presented in this paper was developed for autonomous flower density mapping, with the aim of providing an input into a decision support tool that can in turn be actioned by a variable rate sprayer. The flower segmentation algorithm was based on an improved FCN model which refines segment edges for small apple flowers. To verify the robustness and accuracy of our model, the evaluation was based on both daytime and night-time data captured on a moving vehicle at approximately 2 frames per second in a large commercial orchard. Both can achieve relatively high F_1 score on low resolution images with fast inference speed. Finally, flower density mapping was generated based on detection results by using GPS information for each image.

3. Materials and methods

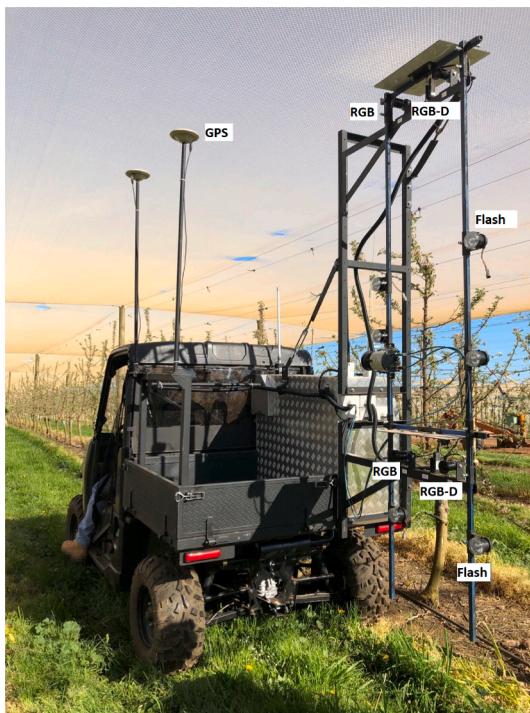
3.1. Data collection

All data were acquired from a commercial apple orchard located in Buttunga, Victoria, Australia, shown in Fig. 1a on 3D trees with tree spacings of approximately 1.5 m and row spacing of approximately 3.5 m. Our study row was of the Gala variety and consisted of approximately 130 trees. Datasets were collected at the later peak bloom stage where some petals had fallen. All night and day images were taken for both sides of the row on the same day in order to verify the consistency of flower density.

For this application, a subset of sensors mounted on a mobile platform built by SwarmFarm in 2018, shown in Fig. 1b was utilised. Two imaging systems were used for daytime collection and night-time collection as shown in Table 1. The night-time system consisted of two Teledyne RGB cameras mounted perpendicular to the row at 1.95 metres and 2.85 metres from the ground in order to capture full canopy. The daytime system consisted of two Intel Realsense D435 RGB-D cameras, mounted the same heights as the two Teledyne cameras. All cameras are approximately 2 megapixels and night-time operation was illuminated by constant halogen lights. The vehicle travelled forward at approximately 5 km/h with cameras facing to the left to capture images



(a) Study row shown as blue line.



(b) Data collection system developed by SwarmFarm Robotics

Fig. 1. Whole farm and the mobile platform.

of apple trees at 1-2 images per second. RTK-GPS is used for positioning and the system includes a computer for data logging. All datasets are summarized in [Table 2](#).

3.2. Ground truth

The detection of apple flowers is formulated as a semantic segmentation task with two classes: ‘flower’ and ‘background’. For this

Table 1
Description of the two imaging systems.

	Intel Realsense D435	Teledyne
Timing	Day	Night
Illumination	N/A	Flash lamps
Type	RGB-D	RGB
Resolution	Color: 1920 × 1080 Depth: 1280 × 720 pixels	Color: 1936 × 1216 Depth: NA
Vehicle Speed	5 km/h	5 km/h

Table 2
Dataset of North-South oriented row.

Date	Phenology	Type	Side	Time of Day
2018-11-01	Later full bloom	RGB-D	Eastern	13:41
			Western	13:49
	RGB	Eastern	21:25	
			Western	21:26

paper, we manually annotated 100 images for the daytime dataset captured from the Eastern side and 100 images for the night-time dataset captured from the Eastern side using the Image Labeler in the Image Processing and Computer Vision toolbox in Matlab 2019a. Pixels of apple flowers manually labeled as the ‘flower’ class are indicated as white in [Fig. 2](#), with the remaining pixels forming the ‘background’ class. Most existing annotation tools use polygons to do coarse labeling but this would be very inaccurate for small objects with complex contours such as those in apple flowers. In this work, we use ‘flood fill’ in the toolbox and make adjustments to represent the real flower contours as well as possible; zoomed examples can be seen in [Fig. 2c](#) and d. Note here we carefully make sure the 100 images within each dataset have no overlap and are as different as possible, as we want the model to be able to cover all cases in field.

3.3. Flower segmentation

The proposed network architecture was adjusted from current state of art semantic segmentation approaches to do flower segmentation, end-to-end, pixel-to-pixel. Training was based on different configurations to investigate the network performance. Evaluation metrics for semantic segmentation are also introduced in this section.

3.3.1. FCNs network

Fully Convolutional Networks (FCN) are one of the most successful networks for semantic segmentation ([Long et al., 2015](#); [Liu et al., 2017a](#)), providing an end-to-end, pixel-to-pixel training for pixel-based classification. The network first goes through the downsampling path to feature interpretation, and then passes through the upsampling path to recover spatial information, shown in [Fig. 3](#). Furthermore, to fully recover spatial location information lost in the downsampling path, the network normally fuses the output from shallow pooling layers which contains finer strides with the output from deconvolution layers. This combination between fine layers and coarse layers helps the model make local predictions that respect global structure. The original research described multiple FCN combinations ([Long et al., 2015](#)), but FCN8s presented the best performance for the public dataset PASCAL VOC. As shown in [Fig. 3](#), two combinations were carried out in FCN8s, where pooling layer 3 (P3) is added to deconvolutional layer 1 (D1) and pooling layer 4 (P4) is added to the deconvolutional layer 2 (D2). Apart from two points above, we also utilized trainable deconvolution instead of fixed bilinear interpolation in original FCN paper ([Long et al., 2015](#)), as trainable deconvolution has been shown to have better performance by [Badrinarayanan et al. \(2017\)](#).

The objective of this paper is to segment apple flowers, which are much smaller objects compared with objects in PASCAL VOC. Previous



Fig. 2. Image Annotation Example where (c) and (d) are the labeled regions for the respective original images (a) and (b). The white colored regions in the annotation represents ‘flower’ pixels and the black colored regions represent ‘background’ pixels.

work has shown that although FCN8s performs well for pixel based segmentation on high resolution objects, performance is noticeably degraded at lower object resolutions (Ma et al., 2020). Therefore, given the low resolution of the apple flower objects in our dataset, a novel network called FCNs-Edge was proposed, as shown in Fig. 3. The

proposed network has two major adjustments based on FCN8s. The first adjustment is to add two more combinations, which are combining pooling layers P1 and P2 with deconvolutional layers D3 and D4 respectively. This is because pooling layer 1 (P1) and pooling layer 2 (P2) contain more local and low-level features such as edges and curves,

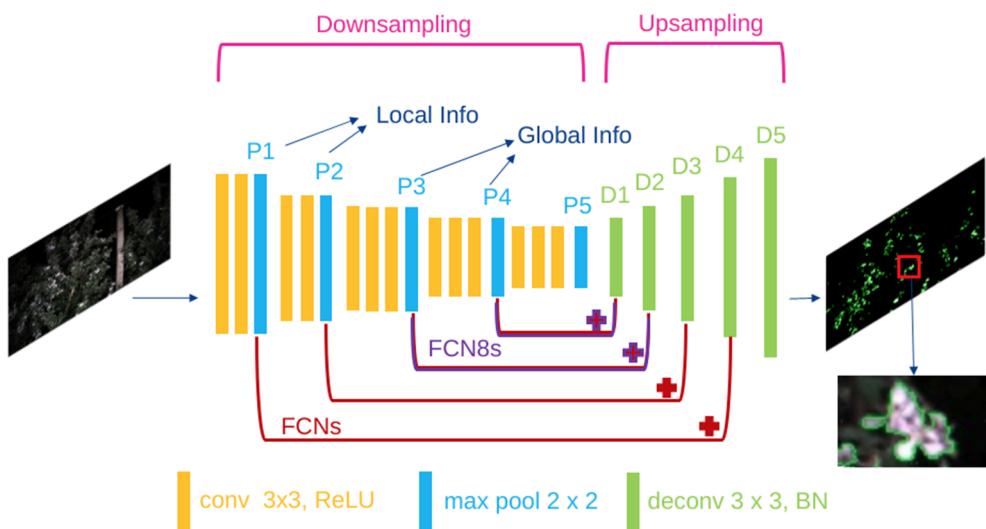


Fig. 3. Pipeline of the FCNs-Edge method.

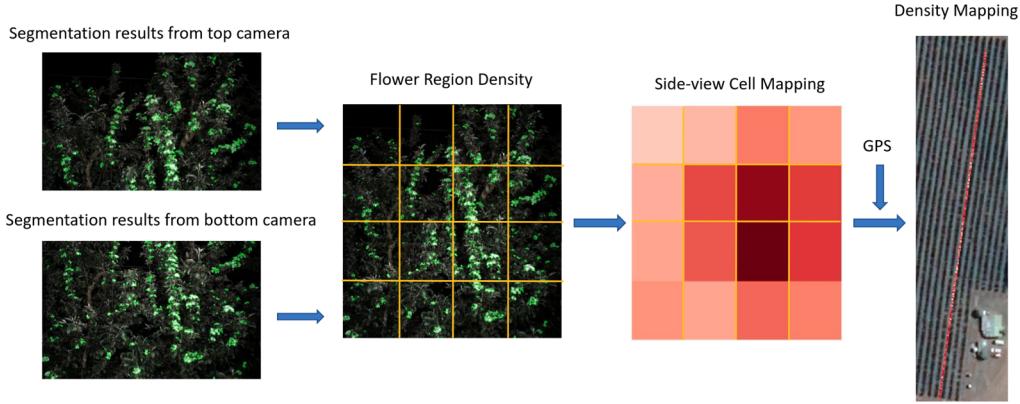


Fig. 4. Apple flower density mapping flow chart.

which can help refine flower contour segmentation (Wu et al., 2018). Another improvement is adding one more edge class artificially during the training, which was inspired from Zabawa et al. (2019), Wu et al. (2018). The class ‘edge’ was generated simply using morphological operations on ground truth masks, to extract pixels around the edges of flowers. The final prediction results have three classes; ‘inner flower’, ‘edge’ and ‘background’. By adding this ‘edge’ class, we intentionally emphasise the importance of flower contours during the training. The network will put more effort into segmenting flower edges, thus refining flower shape.

In addition, the proposed network only consists of locally connected layers including convolution, pooling and deconvolution, without using fully connected layers. This greatly reduces the number of parameters and computation time required when compared to fully connected layers which require a linear operation for each connection in which every input is connected to every output by a weight. The simplicity of the proposed network means that training can be completed in under 100 epochs, which is considered to be very fast for training.

3.3.2. Cost function

For the task of image segmentation, the loss computes pixel-wise cost between prediction maps and ground truth, which is a pixel-wise cross entropy loss. We use the following formula:

$$L = - \sum_{j=1}^m x_j \log y_j, \quad (1)$$

where m is the number of classes. In our case m is 2, corresponding to flowers and background or 3, where the flower edge class has been added.

3.3.3. Evaluation methodology

The segmentation performance of FCN was evaluated on multiple daytime and night-time datasets. To verify the result of apple flower segmentation, we report the following metrics that are commonly used in semantic segmentation work:

1. F_1 score considers both precision and recall, where an F_1 score reaches its best value at 1 and worst at 0.

$$\text{recall} = \frac{t_p}{t_p + f_n}, \quad (2)$$

$$\text{precision} = \frac{t_p}{t_p + f_p}, \quad (3)$$

$$F_1 = \frac{2 * \text{recall} * \text{precision}}{\text{recall} + \text{precision}}, \quad (4)$$

2. Intersection over Union (IoU) is considered the harshest metric among all because of its sensitivity towards results with a high false positive f_p rate or false negative rate f_n . One thing to note that IoU in this paper is on a pixel level basis rather than bounding box basis, calculated as follow:

$$\text{IoU} = \frac{t_p}{t_p + f_p + f_n} \quad (5)$$

Note in all equations that t_p , f_p , f_n denote true positive, false positive and false negative respectively. Classes ‘edge’ and ‘inner flower’ were combined and considered as one class ‘flower’ for the evaluation of the networks. This is because whether the network is able to separate ‘edge’ class from all flower pixels is not important, as the aim of this paper is to segment all flower pixels, which are finally fed into the mapping process.

3.4. Training

During training, two datasets were split into training and test sets. The daytime and night-time images were assigned randomly to the training and test set in a 70% and 30% split respectively. For the daytime only model, daytime images were assigned randomly into training-day and testing-day datasets in a 70% and 30% split respectively. A similar 70%/30% split was used for the night-time only model consisting of night-time images assigned into training-night and testing-night datasets. Finally, for the day + night model, the training set consisted of all the training-day and training-night images, whereas the test set consisted of the testing-day and testing-night images. During the evaluation, a fair comparison was ensured by using the same subsets for training and testing. However, evaluation on different subsets of training and test sets showed no difference in accuracy. Both datasets present around 15 * 15 pixels per apple flower when using a resolution of 320 * 640 pixels. The deep learning framework used for the implementation of FCN is Pytorch 1.1.0. VGG-16 has been selected as the backbone for FCNs given its accurate results in the original paper (Long et al., 2015). The network was initialized using the weights previously computed for ImageNet. No layer was frozen during training, so all weights could be updated by the training on the apple flower dataset. The training for FCN networks was run on a local machine with a single

NVIDIA GEFORCE 1050 GPU (4 GB memory), an Intel i7 CPU and 16 GB RAM and running Ubuntu 16.04 LTS. Due to GPU memory limitations, only one input image of up to 640×1280 per batch or multiple lower resolution input images per batch could be accommodated. One thing to note is the input size needs to be a multiple of 160 pixels due to the FCN network architecture. The overall training process took approximately 40 minutes for an input size of 320×640 using Stochastic Gradient Descent (SGD) with Nesterov momentum at 0.9, with a batch size of 1, and a learning rate of 0.01.

3.5. Flower density mapping

In this paper, flower density mapping is utilized as a method of portraying variation in flower density of different sections of the apple tree across a tree row or block. Traditional mapping in agriculture adopts a top-down 2D view of entire blocks, mainly due to imagery collected from aerial vehicles or satellites (Tang et al., 2016), or ground-based imagery requiring only variation at different GPS locations (Underwood et al., 2016; Liu et al., 2017b; Wang et al., 2018). However, for apple trees, tree heights of up to 4 meters means that the variation that exists between different sections of the tree caused by factors such as varying shoot lengths and light interception differences (Pallas et al., 2016) is lost in a traditional 2D mapping approach when capturing imagery of the full tree. For a variable rate sprayer, the ability to identify variation within each tree will allow different sections of the tree to be treated with different levels of chemical spray, hence the side-view map with a grid as shown in Fig. 5 was adopted, similar to that used by Hu and Whitty (2019) for mapping apple canopy density at different heights. To map flower density variation, flower density and cell sizes need to be defined and the density correlated with GNSS data.

3.5.1. Definition of flower density and cell sizes

The definition of cell elements can be found in Fig. 6, with red squares representing individual map cells, the blue rectangle representing an image overlaid on the map cells, and the yellow square representing cell coverage of an individual map cell. In this paper, flower density is defined as follows:

$$\text{Flower Density} = \text{Cell Coverage} \times \frac{\sum \text{Flower Pixels}}{\sum \text{Region Pixels}} \quad (6)$$

where:

- Flower Density = value between [0,1] where 0 is no flowers, and 1 is 100% flowers.

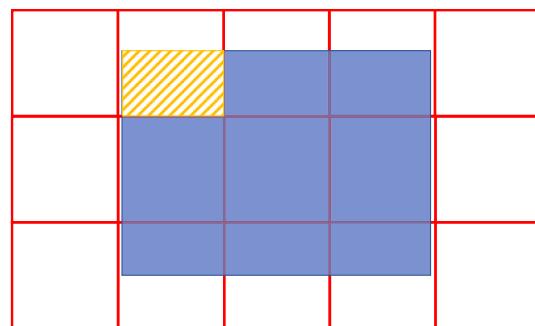


Fig. 6. Cell map representation where red cells represent individual ‘map cells’, the blue rectangle represents an individual image overlaid over the cell map, the yellow rectangle represents the ‘cell coverage’ of the individual image within the second from the left ‘map cell’. Cell bounds are defined by the extent of each ‘map cell’. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- Cell Coverage = percentage of map cell covered by the image.
- Flower Pixels = pixels defined as flowers from the segmentation function within cell bounds.
- Region Pixels = total number of pixels within the cell bounds.

This definition ensures that comparison between cameras or images of differing resolution are consistent.

Cell sizes were defined arbitrarily as a $0.5 \text{ m} \times 0.5 \text{ m}$ grid in a vertical plane along the centreline of the row. From this, the cells can be resampled to lower resolutions to suit the chemical spraying system. In this paper, the $0.5 \text{ m} \times 0.5 \text{ m}$ cells were resampled to $1 \text{ m} \times 0.5 \text{ m}$ grid resolution for display in GIS software that can be exported in a format suitable to the chemical spray system.

3.5.2. GPS-based flower density mapping

GPS was used to assist mapping of flower density (Fig. 1b). The flowchart in Fig. 4 shows where GPS data is utilized in the system, with Fig. 5 intuitively illustrating the density map projection. The offset of the GPS to the cameras are recorded, as well as parameters of field-of-view, end-post locations, and distance to canopy. The GPS information was correlated to the camera information by projecting the camera data into the canopy. The assumption of a constant distance to the canopy is utilised in the absence of LiDAR information. To identify image coverage, the following formula was used for each image dimension



Fig. 5. Density map shown virtually within the rows.

$$\text{Image Coverage} = 2 \times \left(\frac{\sin(0.5 \times \text{FOV})}{\sin(90 - 0.5 \times \text{FOV})} \right) \times d_{\text{canopy}} \quad (7)$$

where:

- FOV is the corresponding horizontal or vertical Field Of View in degrees given for the camera and lens configuration.
- d_{canopy} is the distance from the camera to the apple tree canopy.

Offsets between images were derived from the GPS offset between images and the millimeter per pixel conversion given below.

$$\text{mm pixel conversion} = \frac{\text{Image Coverage}}{\text{Total pixels along dimension}} \quad (8)$$

Finally, the 2D output of flower density was projected geographically using the GPS coordinates of the end-posts for visualisation in GIS software which can then be further utilised for chemical spray applications. The flow chart is shown in Fig. 4.

4. Results

4.1. Flower segmentation

To verify our method, we first performed experiments to compare our proposed algorithm FCNs-Edge with FCNs and FCN8s for both night-time and daytime datasets with the resolution at 320 * 640 scaled from 2 Megapixels. Note here background removal preprocesses for daytime images produced equivalent evaluation metrics to original daytime images, therefore we chose to use original daytime images. From the results shown in Table 3, we can see that FCN8s obtained an F_1 score of 66.2% with a low precision score of 59.6% for the night-time dataset, which means FCN8s has a high false positive rate. The detection overlay shown in Fig. 7a proved that FCN8s can only output rough flower edges with those small gap areas between flower petals also being detected as flower pixels. By combining information from shallow layers of FCNs with information from pool3 and pool4, an F_1 score of 77.8% was achieved and also around a 10% increase in both recall and precision, compared with FCN8s, see Fig. 7b. When adding one more edge class for training the FCNs network, flower edges were further refined as shown in Fig. 7c. This achieved the highest F_1 score of 82.6% among the three models and an increase of 10% in the precision score, compared with FCNs. For the daytime dataset, the situation is the same where an increase by 13% in the F_1 score was achieved by our proposed model compared to FCN8s. FCNs-Edge obtained the highest precision score of 88.3%, an increase of 10% compared to FCNs. Finally, by training and testing on both the daytime and night-time datasets together, an F_1 score of 84.4% was achievable. This proves that training daytime and night-time images separately is not necessary and the proposed algorithm can give good performance even though the two types of images appear very different. The inference speed measured on our GPU is 0.089s per image.

To evaluate further, we tested our proposed model using input images at different resolutions; see Table 4. All tests were on the

daytime and the night-time dataset together as the table shows. Compared to the input size 320 * 640 tested in Table 3, when the input size increased to 480 * 800, the F_1 score slightly increased by 0.8%. However, the F_1 score only increased by 0.3% when the input size further increased to 640 * 1280. Fig. 8 shows FCNs-Edge results on the same image but with different input resolutions, which shows no significant differences among the three resolutions. Note here the green mask is true positive, pink mask is false positive and blue mask is false negative. Fig. 9 shows a night-time whole image example and Fig. 10 shows a daytime whole image example. For presentation, the input resolution of 640 * 1280 was used in these two figures. There are some false positives and false negatives around flower clusters. Those can be attributed to inaccurate labeling due to the difficulty in labeling real edges of complex flower clusters. Instead of inference speed, we also measured the whole process including the time spent reading images, moving tensors between GPU to CPU and generating binary masks for comparison, which is more important for the mapping part. The time spent increased by 0.12s at the resolution of 480 * 800 and by 0.32s at the resolution of 640 * 1280, compared to the time spent for 320 * 640. These results showed increasing the input resolution does not largely increase the segmentation accuracy but increases the time spent per image more than twice. Therefore, for large scale implementation the input size of 320 * 640 pixels is recommended.

4.2. Density mapping

The time taken to generate the density map for a single row on a standard laptop (Intel i5-8250U CPU, 8 GB memory) was 10.8 minutes when using a total of 768 images. Therefore, the time taken to process a single image using the entire pipeline described by Figs. 3 and 4 is 0.7 seconds. Notably, two datasets could be ran in parallel without any significant overhead on the standard laptop, indicating that our pipeline is not resource intensive and therefore suitable for commercial orchards. Examples of the side-view density maps for low, medium and relatively high density are shown below in Figs. 11–13 respectively. Very clearly seen is the variation between the density maps from the eastern and western sides of the canopy. This can be attributed to natural variation in the flowering stage of the trees.

5. Discussion

As seen in Table 3, the proposed FCNs-Edge model for apple flower segmentation outperformed FCN8s and FCNs over a wide range of evaluation metrics. FCN8s was the worse performing of the three due to low precision in identifying edges of the apple flowers as seen in Fig. 7c. Although Ma et al. (2020) was able to achieve an accuracy of 83.7% using a FCN8s based method when identifying ears of winter wheat, they also showed the performance degraded significantly with lower resolution imagery (F_1 of approximately 70%). The imagery used by Ma et al. (2020) contained objects of approximately 50 * 50 pixels at the lowest resolution compared to approximately 15 * 15 pixels for each apple flower in this paper. As the resolution of the objects in our images

Table 3
Summary of results obtained from FCN8s, FCNs and FCNs-Edge.

Dataset	Model	Resolution	IoU	F_1	Precision	Recall	Inference Speed
Night	FCN8s	320 * 640	49.3%	66.2%	59.6%	74.2%	0.089s
	FCNs		63.7%	77.8%	72.6%	83.8%	
	FCNs-Edge		70.4%	82.6%	83.4%	81.9%	
Day	FCN8s	320 * 640	56.7%	72.3%	68.5%	76.7%	0.089s
	FCNs		67.1%	80.3%	78.5%	82.3%	
	FCNs-Edge		74.9%	85.6%	88.3%	83.2%	
Day + Night	FCNs-Edge	320 * 640	73.0%	84.4%	85.9%	82.9%	0.089s

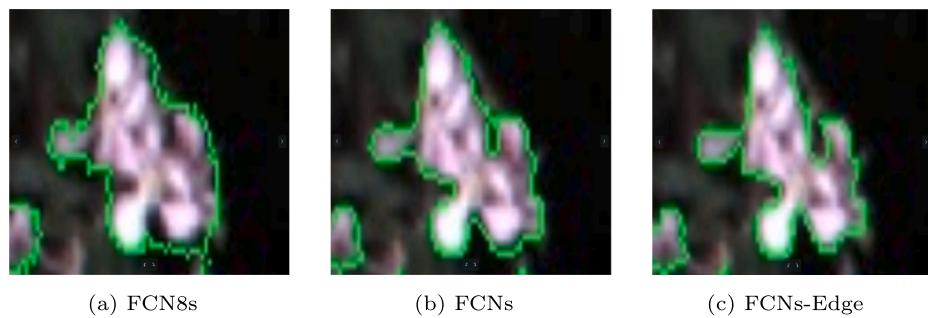


Fig. 7. Selected examples of apple flower segmentation results using different FCN models.

were significantly smaller than those used by Ma *et al.*, the poor performance of the FCN8s method is expected. Therefore, FCN8s is not recommended for detecting objects represented by a small number of pixels. FCNs performed relatively well across our dataset (approximate F_1 score of 79.1%) and has been shown to perform well by Wu *et al.* (2018) for roof segmentation on aerial images. Their testing results all on 2240×2240 images showed a F_1 score of 85.9% using the FCNs-based method and a F_1 score of 73.3% using FCN8s. The input images used by Wu *et al.* (2018) contained objects of approximately 240×240 pixels, which are around 16 times larger than our objects. Their objects are mostly simpler polygon shaped objects than our complex flower shape. Despite the difference mentioned above, this still gave a similar trend to our experiments and proved that FCNs has potential to gain higher accuracy than FCN8s.

One main difference between Wu's method (Wu *et al.*, 2018) and our proposed FCNs-Edge method was that an additional framework in the last layer was created to predict hidden boundaries of buildings caused by shadows or trees. In our case, we only try to predict visible edges of flower clusters as accurately as possible but predicting hidden edges of each flowers is not relevant to our research, hence the multi-task framework was not utilized in our work. However, with such an 'edge' class, their paper showed largely reduced false positive detection based on their detection results, thus it was included in our approach. This also has been proved by a significant increase in precision scores in Table 3. Additionally, their optimized method replacing ReLU with Leaky ReLU reported only an increase of 1% in F_1 score but was at the cost of decreasing the speed of network convergence.

The most relevant work on apple flower segmentation by Dias *et al.* (2018b) presented an approach which was tested using images of approximately 120×120 pixels per apple flower, compared to approximately 15×15 pixels per apple flower for the FCNs-Edge network proposed in this paper. However, our approach was able to achieve an F_1 score of 85.6% in Table 3 relative to their F_1 score of 83.3% both on daytime images, despite the vastly lower resolution. The advantage of using lower resolution was also seen in that the network could fit in a 4 GB GPU, whereas their model was around 11 GB. They also needed to crop sections of the image and feed them in sequentially, and such operations are slow in a traditional CPU-GPU setup, contributing to a slow processing time of 50 s per image, compared with 0.2 s per image for the FCNs-Edge network presented here. Hence our FCNs-Edge network is much more suitable for commercial application than the state-of-the-art published methods.

The impressive performance of the FCNs-Edge model on our dataset

when compared to FCNs as seen in Table 3 can be attributed to improvements to the FCNs model by utilizing local features from shallow layers and creating an edge class. The addition of the edge class in our flower segmentation problem was obtained through basic image processing functions on the labeled images and requires minimal additional effort. It is however noted that when an edge class is utilised to segment different objects for the purpose of counting, such as berries on grape bunches (Zabawa *et al.*, 2019), additional effort in labeling is required to ensure no two objects in contact have the same label. Therefore, the proposed method is recommended where semantic segmentation is already utilised, without delving into instance segmentation. Apple flower density can change drastically within a day due to external factors such as weather (Chmielewski *et al.*, 2011). Therefore, for flower density maps to be actionable in a chemical spray setting, the speed at which a map can be produced is important. Our FCNs-Edge network combined with our flower density mapping pipeline is able to process raw images of one row imaged from two sides (198 metres, 110 trees, 768 image capture points, both top and bottom cameras, vehicle speed at 5 km/h) as well as produce a density map in approximately 10 minutes on a standard laptop with all processes running on the CPU. This means that it is very possible to output a density map for a 1.5-hectare block with 20 rows mentioned above within 3.2 hours after data collection. In contrast, (Dias *et al.*, 2018b) reported 50 seconds for processing of a single apple tree captured from one side of the canopy, which comparatively is 180 minutes for the same one row imaged from two sides and therefore around 60 hours for the entire block. The time spent in terms of block speeds proves the high efficiency of our proposed method. This rapid processing time allows the farmer to make informed decisions in a very short time frame, as the window of opportunity for spraying is weather and resource-dependent and very small, a few days at most.

Light colored netting, backlit images and clouds in the sky are challenging for all methods (including ours) to handle, as they are similarly colored to the white flowers. This could be improved by using the depth channel of RGB-D images to remove sky, where sufficiently robust sensors exist. The appearance of flowers changes dramatically throughout their development stage. This model has only been tested at the late full bloom stage, and further work is necessary to evaluate the effectiveness of the model at different stages of maturity. Future work should also consider adding daytime images with sky and different weather conditions to more fully understand the limitation of the presented method. The processing speed could also be improved by using dedicated hardware or software optimization.

Table 4
Results of FCNs-Edge using input images at different resolutions.

Dataset	Model	Resolution	IoU	F_1	Precision	Recall	Time per Image
Day + Night	FCNs-Edge	320 * 640	73.0%	84.4%	85.9%	82.9%	0.21s
		480 * 800	74.2%	85.2%	85.5%	84.9%	0.33s
		640 * 1280	74.7%	85.5%	86.5%	84.6%	0.53s

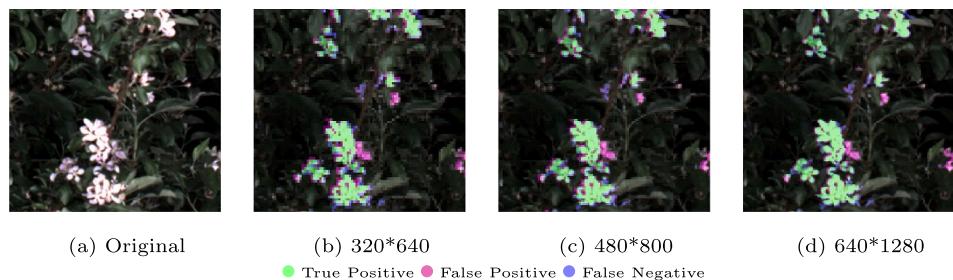
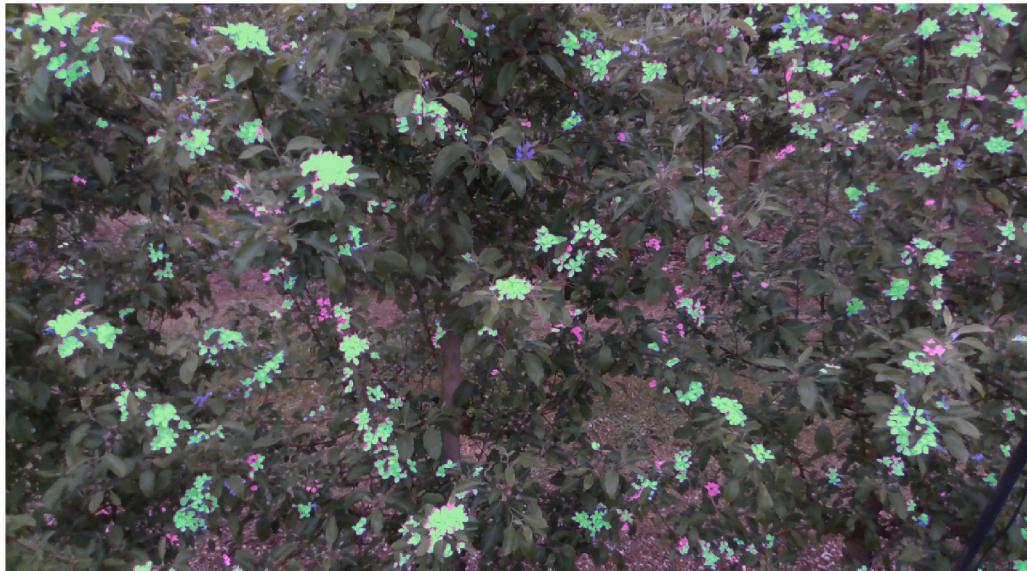


Fig. 8. FCNs-Edge results on the same image with different input resolutions.



(a) Original daytime image



(b) 640*1280 (● True Positive ● False Positive ● False Negative)

Fig. 9. A daytime image example with FCNs-Edge results overlaid.

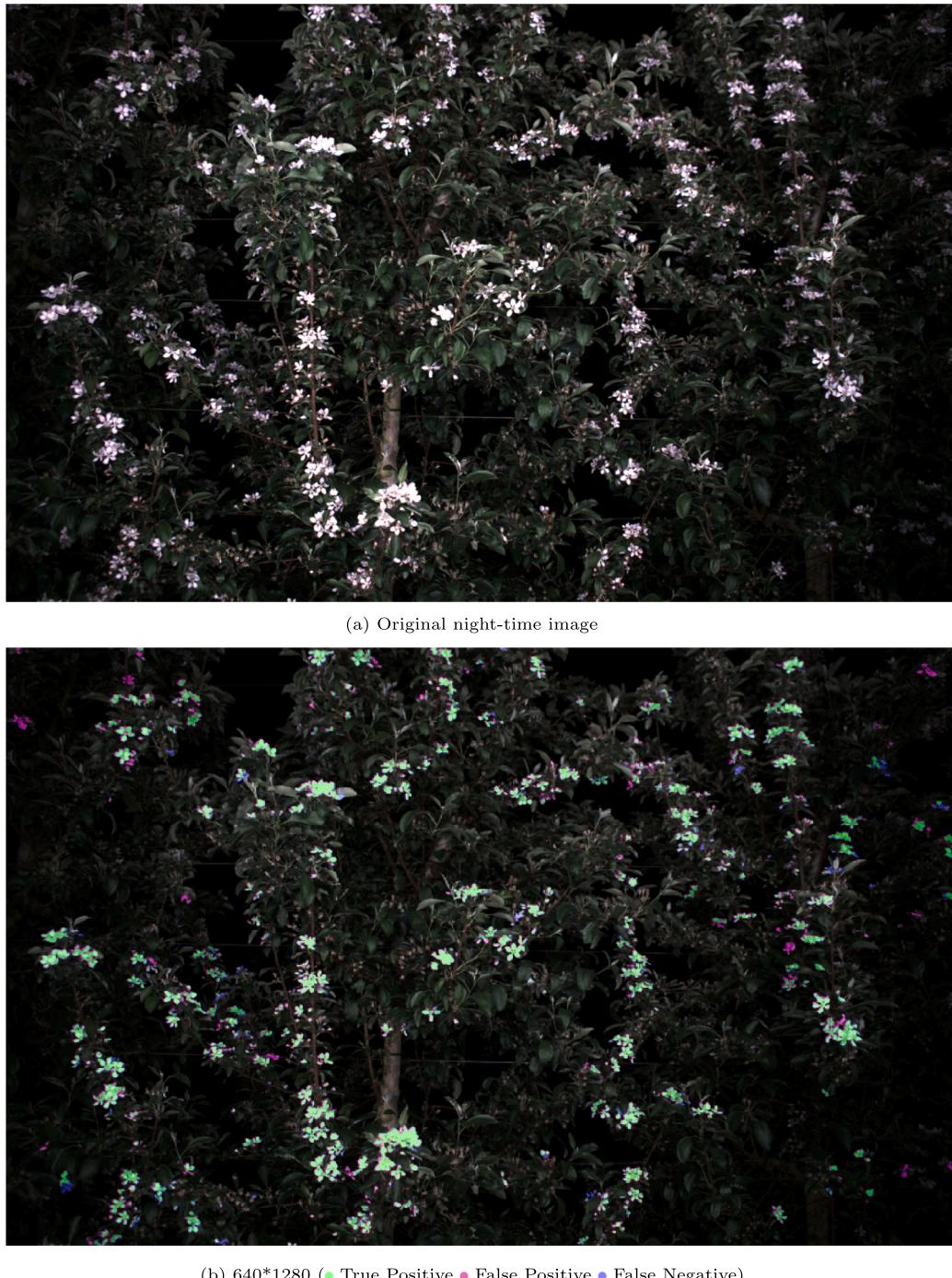


Fig. 10. A night-time image example with FCNs-Edge results overlaid.

6. Conclusion

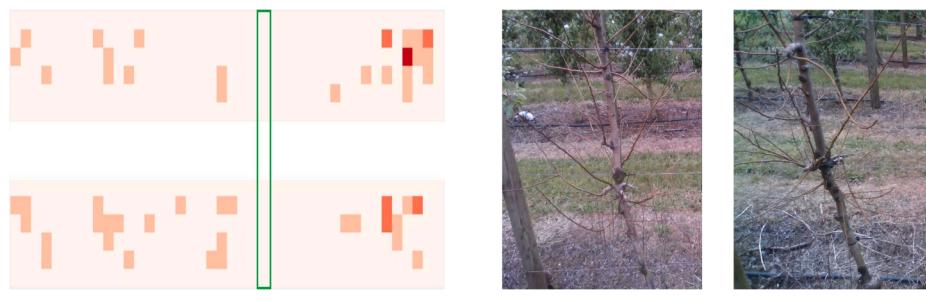
This paper has presented a novel semantic segmentation CNN network ‘FCNs-Edge’ for apple flowers, which achieved relatively high accuracy on low resolution images. The robustness of the proposed method has been proven by testing on both night and day time images which were captured on a moving vehicle under field conditions. The inference speed the proposed network can achieve is considered feasible for the purpose of commercialization. Apple flower segmentation results are then used to generate the side-view density mapping for the purpose of chemical thinning. The proposed method however has potential for application to other small object detection tasks where edges hold critical information for segmentation.

CRediT authorship contribution statement

Xu (Annie) Wang: Conceptualization, Methodology, Software, Validation, Data curation, Writing - original draft. **Julie Tang:** Methodology, Software, Validation, Data curation, Writing - original draft. **Mark Whitty:** Writing - review & editing, Supervision, Project administration, Funding acquisition.

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.



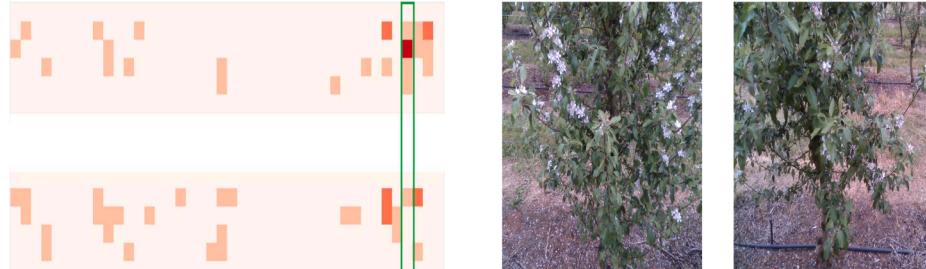
(a) Map comparison from western (top) and eastern (bottom) sides
 (b) Eastern side tree view
 (c) Western side tree view

Fig. 11. Example of output from mapping of daytime data for a section with zero flower density.



(a) Map comparison from western (top) and eastern (bottom) sides
 (b) Eastern side tree view
 (c) Western side tree view

Fig. 12. Example of output from mapping of daytime data with moderate flower density. Some variation in the maps in a) from the two sides (outlined in green) are due to natural variation in the flowering stage of the trees as seen in b) and c). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Map comparison from western (top) and eastern (bottom) sides
 (b) Eastern side tree view
 (c) Western side tree view

Fig. 13. Example of output from mapping of daytime data for a section with relatively high flower density.

Acknowledgments

The authors would like to thank Tom Wyatt, William McCarthy, Tom Brodie, Angus Ross, Angus Hogan from SwarmFarm for building the hardware and collecting ground truth data. Thank you to Hiranya Jayakody and Valerie Mengying Hu for helping build the Google Cloud Platform and data collection. Thank you to the team who did extensive manual labeling. This project has been funded by Horticulture Innovation Australia Limited, using the apple and pear research and development levy and contributions from the Australian Government.

Appendix A

A video to detection and side-view cell mapping results: [Link](#).

References

- Aggelopoulou, K., Wulfsohn, D., Fountas, S., Gemtos, T., Nanos, G., Blackmore, S., 2010. Spatial variation in yield and quality in a small apple orchard. *Precision Agric.* 11 (5), 538–556.
- Aggelopoulou, A., Bochtis, D., Fountas, S., Swain, K.C., Gemtos, T., Nanos, G., 2011. Yield prediction in apple orchards based on image processing. *Precision Agric.* 12 (3), 448–456.
- Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Machine Intell.* 39 (12), 2481–2495.
- Australian Government Department of Agriculture and Water Resources, 2018a. Announcement Information Paper - Commencement of a Review of Biosecurity Import Requirements for Fresh Apple Fruit from the Pacific Northwest States of the United States of America. Accessed online: <https://www.agriculture.gov.au/biosecurity/risk-analysis/plant/apples-usa-2018/announcement-info-paper>.
- Australian Government Department of Agriculture and Water Resources, 2018b. Horticulture: September quarter 2018. In: Agricultural Commodities Report September 2018. Accessed online: <https://www.agriculture.gov.au/sites/default/files/abares/documents/research-topics/ag-commodities/agcommodities-sep-2018.pdf>.
- Bound, S., 2018. Make a smart crop load management choice. In: APAL.
- Chmielewski, F.-M., Blümel, K., Henniges, Y., Blanke, M., Weber, R., Zoth, M., 2011. Phenological models for the beginning of apple blossom in germany. *Meteorol. Z.* 20, 487–496. <https://doi.org/10.1127/0941-2948/2011/0258>.
- Dias, P.A., Tabb, A., Medeiros, H., 2018a. Apple flower detection using deep convolutional networks. *Comput. Ind.* 99, 17–28. <https://doi.org/10.1016/j.compind.2018.03.010>.
- Dias, P.A., Tabb, A., Medeiros, H., 2018b. Multispecies fruit flower detection using a refined semantic segmentation network. *IEEE Robot. Automat. Lett.* 3 (4),

- 3003–3010.
- Dorj, U.-O., Lee, M., ul Imaan, D., 2012. A new method for tangerine tree flower recognition. In: Computer Applications for Bio-technology, Multimedia, and Ubiquitous City. Springer, pp. 49–56.
- Farjon, G., Krikeb, O., Hillel, A.B., Alchanatis, V., 2019. Detection and counting of flowers on apple trees for better chemical thinning decisions. *Precision Agric.* doi:10.1007/s11119-019-09679-1.
- Hočevá, M., Širok, B., Godeša, T., Stopar, M., 2014. Flowering estimation in apple orchards by image analysis. *Precision Agric.* 15 (4), 466–478. <https://doi.org/10.1007/s11119-013-9341-6>.
- Horton, R., Cano, E., Bulanov, D., Fallahi, E., 2017. Peach flower monitoring using aerial multispectral imaging. *J. Imag.* 3 (1), 2.
- Hu, M., Whitty, M., 2019. An evaluation of an apple canopy density mapping system for a variable-rate sprayer. *IFAC-PapersOnLine* 52 (30), 342–348.
- Liu, Y., Minh Nguyen, D., Deligiannis, N., Ding, W., Munteanu, A., 2017. Hourglass-shapenetwork based semantic segmentation for high resolution aerial imagery. *Remote Sens.* 9(6).
- Liu, S., Cossell, S., Tang, J., Dunn, G., Whitty, M., 2017b. A computer vision system for early stage grape yield estimation based on shoot detection. *Comput. Electron. Agric.* 137 (C), 88–101. <https://doi.org/10.1016/j.compag.2017.03.013>.
- Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation, in: In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440.
- Ma, J., Li, Y., Du, K., Zheng, F., Zhang, L., Gong, Z., Jiao, W., 2020. Segmenting ears of winter wheat at flowering stage using digital images and deep learning. *Comput. Electron. Agric.* 168, 105159. <https://doi.org/10.1016/j.compag.2019.105159>.
- Pallas, B., Da Silva, D., Valsesia, P., Yang, W., Guillaume, O., Lauri, P.-E., Vercambre, G., Génard, M., Costes, E., 2016. Simulation of carbon allocation and organ growth variability in apple tree by connecting architectural and source-sink models. *Ann. Botany* 118 (2), 317–330.
- Tang, J., Woods, M., Cossell, S., Liu, S., Whitty, M., 2016. Non-productive vine canopy estimation through proximal and remote sensing. *IFAC-PapersOnLine* 49 (16), 398–403.
- Thorp, K., Dierig, D., 2011. Color image segmentation approach to monitor flowering in lesquerella. *Ind. Crops Prod.* 34, 1150–1159. <https://doi.org/10.1016/j.indcrop.2011.04.002>.
- Underwood, J.P., Hung, C., Whelan, B., Sukkarieh, S., 2016. Mapping almond orchard canopy volume, flowers, fruit and yield using lidar and vision sensors. *Comput. Electron. Agric.* 130, 83–96.
- Wang, Z., Underwood, J., Walsh, K.B., 2018. Machine vision assessment of mango orchard flowering. *Comput. Electron. Agric.* 151, 501–511.
- Williams, M.W., 1979. Chemical Thinning of Apples, John Wiley & Sons, Ltd, pp. 270–300 (Ch. 7). doi:10.1002/9781118060742.ch7.
- Wu, G., Guo, Z., Shi, X., Chen, Q., Xu, Y., Shibasaki, R., Shao, X., 2018. A boundary regulated network for accurate roof segmentation and outline extraction. *Remote Sens.* 10(8). doi:10.3390/rs10081195.
- Zabawa, L., Kicherer, A., Klingbeil, L., Milioto, A., Topfer, R., Kuhlmann, H., Roscher, R., 2019. Detection of single grapevine berries in images using fully convolutional neural networks, in: In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.