Precision Agriculture
https://doi.org/10.1007/s11119-019-09679-1

Detection and counting of flowers on apple trees for better chemical thinning decisions

Guy Farjon[1] · Omri Krikeb[2] · Aharon Bar Hillel[1] · Victor Alchanatis[2]

Abstract

Accurate chemical thinning of apple trees requires estimation of their blooming intensity, and determination of the blooming peak date. Performing this task, as of today, requires human experts to be present in the orchards for the entire blossom period or extrapolate using a single observation. Since experts are rare and in high demand, there is a need to automate this process. The system presented in this paper is able to estimate the blooming intensity and the blooming peak date from a sequence of tree images, with close-to-human accuracy. For this purpose, a two years dataset was collected in 2014–2015, partially tagged for the flowers location and completely annotated for blooming intensity. Using this dataset, an algorithm was developed and trained with three stages: a visual flower detector based on a deep convolutional neural network, followed by a blooming level estimator, and a peak blooming day finding algorithm. Despite the challenging conditions, the trained detector was able to detect flowers on trees with an Average Precision (AP) score of 0.68, which is on a par with contemporary results of other objects in detection benchmarks. The blooming estimator was based on a linear regression component, which used the number of flowers detected and related statistics to estimate the blooming intensity. The Pearson correlation between the algorithm blooming estimation and human judgments of several experts indicated high agreement levels (0.78–0.93) which were similar to the correlations measured among the human experts. Moreover, the developed estimator was relatively stable across multiple years. The developed peak date finding algorithm identified correctly the orchard's blooming peak date, which was used to determine the thinning date in the current practice (the entire orchard is thinned in the same day). Experiments testing the algorithm's ability to find a blooming peak date for each tree independently showed encouraging results, which may lead upon refinement to a more precise practice for tree-specific thinning.

Keywords Precision agriculture · Visual detection · Deep convolutional neural networks · Blooming intensity estimation · Chemical thinning

✉ Guy Farjon
guyfar@post.bgu.ac.il

[1] Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer Sheva, Israel

[2] Institute of Agricultural Engineering, Agricultural Research Organization, The Volcani Center, Bet Dagan, Israel

Published online: 20 August 2019

Introduction

The agricultural sector faces an immense challenge—the world population is exponentially growing, and with it, the demand for agricultural yields. The need to satisfy this demand led to significant progress in precision agriculture applications, using advanced technologies including computer vision, satellite navigation systems, remote sensing, geographic information systems, and many others. This work faces the challenge of estimating blooming intensity of apple trees, a part of the chemical thinning process, using advanced computer vision tools.

In agricultural sciences, thinning refers to the removal of a fraction of the trees' flowers or fruits, to improve the growth of others. The link between thinning and fruits quality has been well established in the literature (Forshey 1986; Link 2000). During their blossom period, apple trees produce many flowers, which later transform into apples. Each such flower requires resource allocation from the tree in order to grow properly and turn into an apple. However, when the number of flowers is too high there is an abundance of apples, which hence grow to be small, with insufficient quality and may be unworthy for sale. When this number is too low, the trees produce a small amount of apples, which are worthy for sale but with a low revenue per acre. In order to get the optimal number of apples, which depends on cultivar, growing site and market price structure, trees must undergo a thinning process. In addition to keeping the optimal amount of fruits on the tree, the result of proper thinning keeps the fruits crops stable for years to come, thus avoiding a phenomenon termed 'biennial bearing'. This phenomenon causes the tree to yield an unstable amount of fruit in a two-year cycle: in the first year, the tree produces an excessive amount of apples. In the next year of the cycle, the tree produces a small number of apples. This phenomenon directly impacts the size and quality of the fruits and avoiding it is important for long term efficiency.

When a tree carries too many fruits, it yields small or low-quality fruits, and in extreme cases, this can lead to branches break down due to the load (Dennis 2000). Therefore, the trees are thinned using one of the following methods:

1. Late Manual Thinning—such thinning is usually performed after the physiological fruit drop. The farmers remove a fraction of the fruits, based on their size and proximity to other apples. This method was already recommended by several early English horticulturists in the seventeenth century and was reported to be useful by Gourley back in 1922 (Gourley 1922), but still remains one of the most widely used methods for thinning today (Dennis 2000). It has two main drawbacks:

    - Manpower—using this method, to thin one acre, there is a need for four to five working days. This has aspects of manpower availability as well as high costs.
    - Timing—Since the manual thinning is carried out late, it has reduced effect on improving the fruit size and quality.

2. Chemical Thinning—Chemical thinning is done by spraying chemical solutions suitable for this purpose. The entire process is accompanied by experts that can evaluate the trees condition based on their physiological states. In each day during the blossom period, the experts examine the trees and give each of them a score in the range (0 to 5), which represents their blooming intensity (where 0 is lowest blooming intensity and 5 is the highest). The purpose of this rating system is to find the peak blooming date of each tree. The blooming peak is defined to be the day before the blooming intensity starts to

> decrease. When the blooming peak is found the thinning date of the entire orchard can be determined. The blooming peak day must be carefully selected. Spraying the chemical solution too early or too late can lead to over thinning causing the trees to produce a small number of apples. Because of these reasons, farmers hesitate to use this method, even though it is considered to be better, cheaper and more efficient than manual thinning.

Chemical thinning is considered one of the most effective methods to improve apple quality, size, and color, and in addition, helps in reducing the biennial bearing phenomena (Yoder et al. 2009). Though chemical thinning in the blossom period is superior to post-bloom thinning, it remains one of the more unpredictable parts of apple production with large variations within years and from year to year (Robinson et al. 2010). Hence, there is a need for a precise system that could overcome different variability issues. Chemical thinning relies on an accurate estimation of the blooming intensity of the trees, as it determined the time of application. As of today, this estimation is done by human experts that visually inspect a limited number of trees at selected time points during the 10 days of the blossom period. Since this process requires the experts' physical presence in the fields, they find it hard to provide full support for all the farmers in need. Vision based estimation of blooming intensity can reduce the dependency on experts and allow automation of this task. With an accurate detection system using computer vision tools, blooming intensity estimation can be achieved using a simple digital camera operated by an unskilled worker.

The objectives of this study were: (a) to build an automated vision-based system which estimates apple tree blooming intensity and supports peak day determination with accuracy matching the human expert's accuracy. The system should be applicable in field conditions and invariant to differences between years; (b) to conduct field test evaluating the system and to quantify the degree to which it achieves the goals stated in (a). The research hypothesis was that such a system can be built based on recent advances in deep networks for vision. The knowledge sought in this study is its construction details and obtained performance. The work was done based on the engineering method guidelines focusing on problem definition, data gathering, system construction, and evaluation. For more information about the engineering method process see (Koen 1985).

Background

There are numerous studies related to the problem of object detection in an agricultural environment, with the aim of automating agriculture tasks which are highly labor intensive (Gongal et al. 2015). Most of the suggested object detection algorithms are based on finding pixels colors from different colors spaces (for example—RGB or HSV). These algorithms mostly involve a binarization of an image using tuned thresholds, which are variant to changes in illumination, camera position etc. For example, Aggelopoulou and his collaborators (Aggelopoulou et al. 2011) first converted the image from RGB to a single channel measuring the distance to a pre-defined flower color. Then, they create a binary image by applying a threshold. Finally, they regressed the yield of the given tree from the number of found pixels. In another example, Adamsen et al. (2000) used image processing tools to isolate areas in the Lesquerella plant images that contain flowers pixels, then estimate the number of the flowers in the image based on an Euler number method. Though their system successfully estimates the number of flowers, it analyzed one image in 3.5 min, which may be a major setback when using a large dataset with hundreds or thousands of images.

In (Hočevar et al. 2014) the authors use image processing tools to estimate the number of flower clusters. Their method starts by converting the image from RGB color space into hue, saturation & lightness (HSL) color space. Then, they created a binary representation of the image using hard thresholding for each of the HSL channels. To avoid noise, they rejected areas in the image that were too small or too large to be considered as flowers clusters, measured by the number of pixels in each cluster. This method suffers from hard-coded parameters, such as the threshold of each channel in the HSL color space. In (Wang et al. 2013; Sa et al. 2016) the authors developed a computer-vision system to detect apples, but only in controlled illumination, where the end task is yield estimation. Their system uses colors of pixels as a key feature to detect apples but suffers from undetected green apples, since they were partially covered by the tree foliage. Linker and his associates (Linker et al. 2012) addresses these specific issues in their work. First, they performed their research under natural illumination conditions and second, they quantify the number of green apples in RGB images. Though they were able to successfully find ~ 85% of the green apples, they still encountered a large number of false positive detections.

In recent years, deep Convolutional Neural Networks (CNNs) have significantly improved the accuracy in computer vision tasks and are widely used in several classic vision tasks as object classification (Krizhevsky et al. 2012), detection (Ren et al. 2015), and segmentation (Long et al. 2015). These networks are better able to cope with real-world vision problems than previous technologies, hence providing new automation opportunities. Object detection in images is considered a hard task, even on the benchmark datasets PASCAL—VOC (Everingham et al. 2010) which has received substantial attention in recent years. Detection in real outdoor settings, rather than in a controlled one, make this task more difficult (see the discussion in the data acquisition section). When a CNN is applied to an image, the image is processed with a number of convolutional layers, each composed of multiple convolution operations and a non-linear operator. The output of this convolutional processing is a high dimensional feature map. In such a feature map, each location in the original image is represented as a column of features describing the area around it. During training, the parameters of the convolutional operation, termed filters, are gradually modified to minimize a target loss of interest using a gradient descent optimization procedure. In the resulting model, higher layers in the network contain semantic features, providing information which enables decision regarding object identity.

In the case of agriculture, the environmental conditions present several challenges for computer vision tasks such as illumination variation, object occlusion, and the large internal variance of the flower object class with respect to size, appearance and posture. In similar tasks, CNNs were found to be useful since they avoid the need for hand-engineered features: with enough data, they learn good representation and provide a system with high resistance to irrelevant variability issues. Such networks were shown to provide state-of-the-art results in the agricultural domain for detection of fruits in harvesting robots (Potena et al. 2016; Sa 2016), disease identification in crops (Mohanty et al. 2016) and more.

Specifically, Bargoti and Underwood (2017) used the faster R-CNN algorithm (Ren et al. 2015) to detect mangos, apples, and almonds. Though they succeed to detect apple and mangos with good results, they struggle to detect the almonds on the tree, since they are smaller and harder to notice. The latter is close to the problem of detecting flowers on an apple tree, since these small objects often overlap with one-another and are harder to track. Another example can be found in (Sa et al. 2016) work, in which they used the faster R-CNN algorithm to detect sweet peppers and rock-melons in RGB and Near-Infrared

(NIR) images, for harvesting purposes. This work also demonstrates the success of faster R-CNN and its ability to detect objects in an agricultural environment. A noticeable difference between flower detection (presented in this paper) and fruit detection is that fruits usually grow apart and far from one another while flowers (and in particular apple flowers) grow one next to each other in a cluster formation. This work follows a similar approaches to in (Sa et al. 2016; Bargoti and Underwood 2017) for flower detection, but uses the information for regression of the blooming intensity level and determination of peak blooming day. Another approach to count objects in an image is to use a CNN as a regressor. Meaning that instead of detecting the flowers (and their position), this approach takes an image as an input and directly regress the number of objects in it. Dobrescu and his collaborators (Dobrescu et al. 2017) uses this approach to count the number of leaves of the Arabidopsis and Tabaco plants dataset, published by (Minervini et al. 2016). Though the Arabidopsis and Tabaco plant images in the datasets shows that the leaves overlap, which resembles to the issue this work as encountered with, they usually contain a relatively small number of leaves on a single image-centered plant.

Development of decision support systems for chemical thinning is a limited area of research in the professional literature. In (Robinson et al. 2010), the authors present predictive system that can help growers understand when to chemically thin the apple trees and with which solutions. In their research, the authors try to assess the effect of different chemical solutions on three kinds of apples—'Royal Gala', 'McIntosh' and 'Ace Delicious'. This assessment is done by a prediction model based on location-specific measurements as temperature and sunlight. This research does not consider blooming intensity estimation, but only examines the effect of different chemical solutions, sprayed at different time-windows, on trees yield. This system can only be used when the parameters of the model are known (tuning the parameters took a decade in this specific research) and is applicable only if the peak date is already known to the apple growers.

The system presented in this paper was developed for blooming intensity estimation based on a CNN flower detector, and was tested for performance by comparing it to human expert judgments in real field conditions. Data was collected from two consecutive seasons and annotated for both flower positions and tree blooming intensity. A robust flower detector was successfully trained and estimated for detection performance. Based on the trained detection system, a blooming intensity estimator, which is close to human level accuracy, was built, and carefully measured for performance. Finally, the capability of the resulting system for determining the day of the blooming peak was estimated and analyzed.

Materials and methods

Data acquisition

The data for this work was acquired in Matityahu farm in the Western Galilee area in Israel (33° 4′ 0.28″ N 35° 27′ 8.97″ E), at altitude of 680 m above sea level. At 2014, 60 trees were tested between the 4th and the 9th of April and in 2015, 159 trees were tested between the 8th and the 15th of April. During this period RGB images of each tree were acquired daily resulting in 300 and 795 images in 2014 and 2015 respectively. The examined trees were of the Golden Delicious apple kind and were between 1.9 to 3 m tall. The distance between tree rows was 3 m and the interval between trees within the row was 1.5 m. All images were taken using a Canon 6D camera (20.2 megapixels) equipped with Canon's

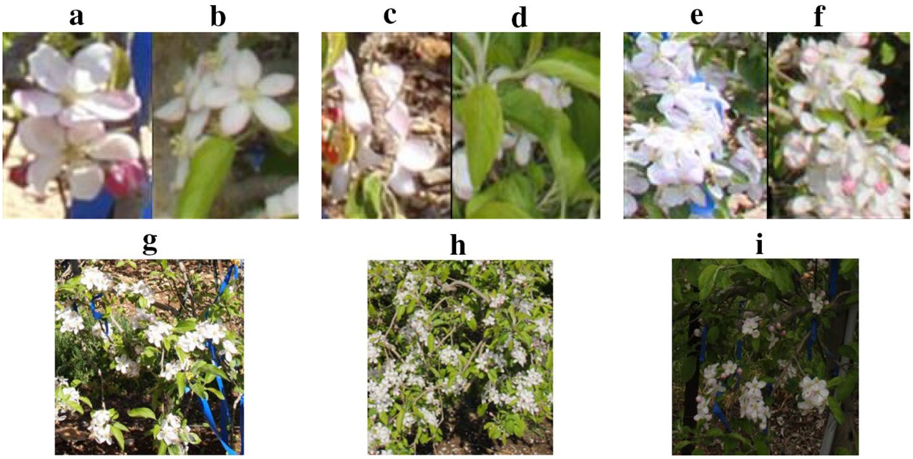

Fig. 1 Variability difficulties of the flower detection task. **a**, **b** viewpoint variations. A contains frontal flowers while B contains some flowers from a lateral view. **c**, **d** partial occlusion. In **c** a branch hides two flowers and in **d** leaves cover a group of flowers. **e**, **f** clusters. The apple flowers grow in tight groups, which sometimes make it almost impossible to separate and count them. **g**, **h** Scale variability. This images were cut using the same rectangle size. As can be seen, flowers in **g** are significantly larger. **i** Illumination variability. Some images and image regions are significantly darker than others. While **g**, **h** contain flowers which are almost absolute white, flowers in **i** have a pink hue (Color figure online)

24 mm prime lens (at 2015, Polaroid filter was added on the lens to prevent specular reflectance), between the hours 09:00 AM and 12:30 PM. The camera was placed on a tripod, in approximately 1.5 m away from the trees. Images were saved in a.jpg format in resolution of 3648 × 5472 pixels (width × height).

This paper presents two main tasks: detection of flowers on the trees, and estimation of blooming intensity. For the detection task, 20 images from both 2014 and 2015 were selected (10 images from each year) and the bounding box around each flower was manually marked. Apple flowers grow in clusters, where each cluster can potentially contain up to 6 flowers which are in high proximity to one another (see in Fig. 1c). This high proximity required fine and careful labeling, separating between partially occluding flowers. MATLAB's image labeling tool (Image Training Labeler, introduced in version 2014a) was used for this purpose. In total, 2893 flowers were marked and served as positive set of flower samples, while negative samples were mined from the non-marked area of the same 20 images.

For the task of blooming intensity estimation, all the trees (159 trees from 2015, and 60 from 2014) were annotated by an expert at one time point each year, close to the estimated time of peak. A small subset of the dataset containing 62 tree images, taken in a single day (the 13th of April, 2015), was annotated for blooming intensity estimation by a second expert. This subset was used for checking the agreement between two human estimators, and for checking the agreement of the algorithmic estimation with each of them is a single-day.

The full dataset, is considerably larger, it included images and annotations in several days, from the beginning of the blooming period, until the day after the peak. Its annotation was done by the less trained experts. Blooming peak date was estimated in a sequential manner: the blooming intensity in a certain day was compared with the blooming intensity of the same tree in the previous days, in order to determine the peak blooming date. All the blooming intensity annotations were done based on observation of the trees only from one side, the side that faced the sun, as this is the practice used by thinning experts. In order to develop a generalized model, images from a wide range of environmental conditions were acquired (Kapach et al. 2012; Gongal et al. 2015):

- Images acquired under different illumination conditions were chosen to cover illumination variations due to weather or camera characteristics (like shutter speed).
- Partially occluded flowers were tagged as flowers to enable better detection with partial occlusion.
- Different flower scales and postures were considered to cover object posture variations
- Sufficient images from each blooming intensity class were annotated.

Figure 1 depicts examples of flowers in the acquired images, showing the great variability in the data set.

Processing algorithms

The developed system included three components: a flower detector, a blooming intensity estimator, and a peak-day finding algorithm.

For flower detection, the faster R-CNN detector presented by Ren et al. (2015) was used as a base algorithm. Changes were made to both the algorithm and the dataset to adapt the algorithm to the specific environmental and imagery issues.

Based on the flower detector, two models for blooming intensity estimation of apple trees were built, when two different scenarios were considered. In the first scenario, estimation of the blooming intensity was done as an isolated event, based solely on a single image of the tree at a single time point. In this task, the inference was a plain image-to-estimate task, hence was termed on-sight estimation. In the second scenario, blooming intensity estimation was done for in a sequential context, as part of a series of estimations of the same tree across several days. This is a more complex task, in which the system tried to mimic the context of the human blooming intensity decision, based not only on the current tree state but also on its state in previous days.

As discussed above, the goal of blooming intensity estimation was to determine the blooming peak date, since it directly affects the time for application of the chemical thinning. In current agricultural practice, the blooming peak day is defined as the day when 80% of the trees in the entire orchard have reached their blooming peak. The developed algorithms were used to estimate the blooming peak of each tree and hence make a decision for the whole orchard. Nevertheless, it can also be used to make tree specific decisions and apply variable rate spraying.

Flower detection using faster R-CNN

To detect the apple flowers the faster R-CNN architecture (Ren et al. 2015) was used. The faster R-CNN detection algorithm does not train the network from scratch. The detection dataset is small, thus the training stage was done by using a pre-trained classification architecture which was trained on a larger dataset. The implementation in this work used the VGG-16 network implementation (Simonyan and Zisserman 2014), pre-trained on the ImageNet dataset (Deng et al. 2009).

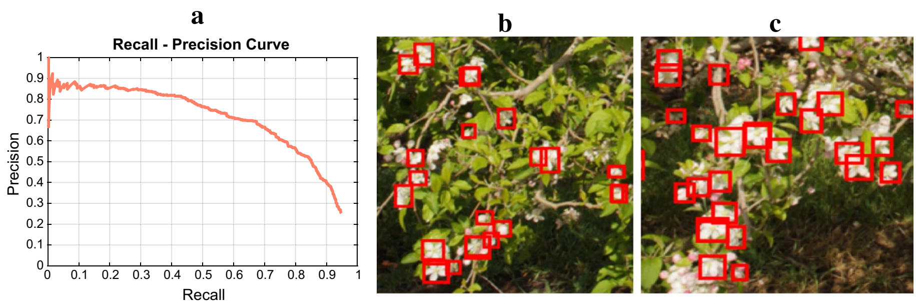

Fig. 2 **a** A recall-precision curve for flower detection, obtained on the 5 test images containing 819 flowers in total. The threshold for positive detection of a flower was set to IoU > 0.3 (see data collection section). The detection and false alarm rates were comparable to those obtained on PASCAL—VOC dataset (AP = 0.683), but with lower localization accuracy (as there IoU > 0.3 threshold was used). This setting fits the blooming intensity estimation application well, as exact flower localization was not of interest for it. **b**, **c** algorithm results on typical two sub images. Each tree image contained 49 such sub images

The detection task was decomposed into two stages (see Fig. 2), implemented as two different but connected network modules. The first network module was the Region Proposal Network (RPN), responsible for finding areas in the image that are likely to contain an object, called Regions of Interest (ROI). This module identified rectangles which may contain objects based on general considerations and was not specific to a certain class. This module had a high false alarm rate but a low misdetection rate, meaning that it detected a lot (thousands) of rectangles in an image, including most of the objects but many non-objects as well. The selected ROIs were then moved into the second network module, which classified them into M classes + background (for negative examples), but also further refined the suggested bounding boxes using a finer box-regressor, trained to minimize an object-specific loss function. This classifier was trained to discriminate flower versus background. The presented implementation started by cropping the original image into sub-images (see cropping method below for details). During *testing*, the RPN returned $N_p = 500$ bounding boxes per sub-image, which is significantly higher than the original number used in Ren et al. (2015). This is because the number of flowers in each sub-image is usually much larger than the number of objects in a typical PASCAL—VOC image. Each bounding box received a score within the range [0, 1], depicting the confidence level of the algorithm regarding flower presence in the ROI. When this confidence level exceeded a pre-defined detection threshold, the algorithm declared that a flower was found.

Since the original images were very large (3648 × 5472 pixels), and the size of the flowers was small (70 × 100 in average) some algorithmic details had to be modified to enable detection of many small objects instead of few large ones. Hence, a cropping method was developed to crop the images into $j \times j$ parts, with an additional small padding on the right and bottom sides to avoid losing information about the flowers positions. These j^2 sub images were subjected to the detection algorithm and detected flowers were then embedded back to their position in the large image. Various j values were tested and $j = 7$ was selected since this division provided best empirical results.

Implementation details

The faster R-CNN algorithm was built based on the PASCAL—VOC dataset properties, some of which were unsuitable for the flower detection task. In addition to the image size issues stated above, several other modifications were made:

Anchors sizes—The anchor parameters are a part of the RPN network, stating the initial area sizes that might contain objects. These initial sizes were refined by the RPN to obtain object proposals. For the Pascal dataset, the anchors were initiated as boxes with areas of 128^2, 256^2 and 512^2 and aspect ratios of 1:1, 1:2 and 2:1 which were too big for the relatively small sized flowers. Hence, the anchors were adjusted to suit the flowers' sizes and were initiated to be areas of 32^2, 64^2 and 128^2 and aspect ratios of 0.75:1, 0.9:1 and 1:1. These ratios were selected by manual and visual exploration and found to give good results.

Percentage of positive and negative examples—As mentioned above, the RPN extracted ROIs from the image. Each of those ROIs received a score (within the range [0, 1]) representing the confidence of containing an object. During *training*, the RPN extracted $N_p > 6000$ bounding-box proposals from each sub-image. These ROIs usually overlapped, hence a Non Maxima Suppression (Hosang et al. 2017) procedure with threshold at 0.7 was applied to remove redundant rectangles. At the end of the process, the 2000 proposals with the highest confidence of containing an object were selected. To decide whether a detection hypothesis should be labeled as object or background, the metric termed Intersection over Union (IoU) was used. The metric was computed using two rectangles: the detection rectangle and a ground truth object rectangle. It was calculated by dividing the area of overlap between the two rectangles by the area of their union. When the IoU between an object detection hypothesis and the ground truth was $\mathrm{IoU} > 0.5$ the bounding-box was considered as foreground (flower) and when it was $0.1 < \mathrm{IoU} < 0.5$ the bounding-box was considered as background. Bounding-boxes with $\mathrm{IOU} < 0.1$ were discarded since they considered as easy negative examples. The ratio between positive and negative examples was set to 1:1 since the number of flowers in each sub-image was relatively high (often between 5 and 30).

Tolerance to pose deviation—Usually in detection tasks, the objective is to find the exact position of the objects in a given image. Given this task, in order for a detection hypothesis to be declared as "hit" the IoU between the ground truth and the detection hypothesis should be greater than $t = 0.5$. In this work the detection was an intermediate sub-task, where the goal was to estimate the blooming intensity, and the exact position of the flowers was of low importance: the system needed to count them, not to localize them accurately. Hence this parameter was set to a lower value of $t = 0.3$.

Blooming intensity estimation

Based on the results of the flower detection algorithm, the tree's blooming intensity was estimated, on a similar scale that the human expert scores the trees, between 0 and 5. In order to trace the experts' estimation, two different linear regression models were used—an on-sight model and a sequential model.

On-sight estimator: The on-sight estimator estimated the tree blooming intensity based on a single image, with no past information about the tree. This model included four explanatory variables:

1. *Number of flowers*—this number was the output of the detection phase with a detection confidence threshold set to 0.83. The detection threshold parameter was tuned during estimator training.
2. Number of flowers squared
3. *Average flowers size*—this variable was chosen based on the assumption that larger and more mature flowers indicate advanced stage of blooming, whereas smaller flowers indicate that the tree is still in an earlier blooming stage.
4. *Number of flowers for additional detection thresholds*—the selected detection thresholds were 0.7, 0.75, 0.8, 0.92, and 0.99.

Sequence-based estimator: This estimator took into consideration time series features describing the tree in previous days. Observing the expert annotator decision making, such features were shown to implicitly or explicitly affect his blooming level estimation. Thus in this model, the variables presented below were added to the variables of the on-sight estimator:

1. The day index in which the image was taken—as stated above, the dataset contains images of the trees for 5 days during the blooming season (2015). Thus, the day index in {1,…,5} was used as a feature. The motivation was that the expert expects blooming intensity increase with day index.
2. Difference between the numbers of flowers in last two consecutive days—this variable tried to capture the current trend of blooming intensity (increase or decrease).
3. Difference (between the numbers of flowers in last two consecutive days) squared
4. Difference between the current number of flowers and the average number of flowers on the tree in previous days squared

Blooming peak date estimator

After estimation of the blooming intensity for each tree, in each day, the final task was to determine the blooming peak date of the tree, and of the entire orchard. The blooming peak estimator was built to enable real time decision making, hence the decision for day i can only rely on information gathered at days 1,…,i. The algorithm mimicked the expert's logic in a very simple manner. For a specific tree, it detected the first day in which the blooming intensity started to decrease, and set the blooming peak to be the day before. To reduce the effect of measurement noise, a decrease was only declared if $S_i - S_{i-1} > \varepsilon$ for some $\varepsilon > 0$, with S_i the blooming intensity of day i. This threshold (ε) was related to the reliability level of the estimation: when it was low ($0 < \varepsilon < 0.5$), the reliability attributed to the blooming intensity estimator/annotator was relatively high (so if the system observed a decline of 0.5 rank, it determined the peak date) and vice versa when it was high. If no blooming decrease was detected across the entire day's sequence, the algorithm declared the last day as the peak blooming day.

In order to determine the orchard's global peak blooming day, the fraction of trees that had already reached their peak was accumulated for each day, building a Cumulative Density Function (CDF) of the peak reaching probability. When the fraction of trees which have reached their blooming peak first reached 0.8 or above, that day was declared as the global blooming peak date of the orchard.

In order to compare human and algorithmic decision regarding the peak blooming day, the procedure for determination of the blooming peak date was applied twice: once for the human blooming annotation data and once using the estimations provided by the blooming estimation algorithm (using both the on-sight and sequenced-based estimators).

Other processing algorithms

Most of the methods found in the literature (Aggelopoulou et al. 2011; Adamsen et al. 2000; Hočevar et al. 2014; Wang et al. 2013; Linker et al. 2012), are based on pixels of a specific color using image processing tools. Two such alternatives were implemented, and compared to the proposed model in this work, for each of the years 2014, 2015.

For each year, the typical flower color was determined using the same ten images that the detector used in the training stage. Both RGB and HSV color spaces were considered. From each of the 10 images, 5 randomly flower bounding-boxes were selected and resized to $65 \times 65 \times 3$, which was the average flower size in the dataset. The target flower color was calculated from these flowers as the median value of each channel (since the bounding-boxes also contains some background information). For color-based flower detection, a binary image was computed by thresholding the RGB values. Several thresholds were examined, finally choosing the one which maximizes the system's accuracy. In the first implemented model, the number of pixels found in the selected color range was used as an explanatory variable, and blossom intensity levels were regressed directly from it. This method is referred to as 'baseline 1'. In the second model, the method presented in (Adamsen et al. 2000) is implemented—the binary image is used to compute the Euler number of the image and used that number as the explanatory variable. This method is referred to as 'baseline 2'.

Algorithms evaluation procedures and performance measures

The annotated detection dataset, which contained 20 tree images, was divided into 2 parts: 15 images for training the detection model, containing 2074 flowers, and 5 images containing 819 flowers for testing it. Both train and test sets contained images from 2014 and 2015 of apple trees. Note that despite having a small number of images hundreds of flowers and many thousands of non-flower rectangles were contained in those images, providing an effective sample size for testing.

In order to evaluate the detection performance, recall and precision indicators were used, defined by:

$$(1) \quad recall = \frac{True\,Positive}{True\,Positive + False\,Negative}$$

$$(2) \quad precision = \frac{True\,Positive}{True\,Positive + False\,Positive}$$

Changing the confidence threshold of the algorithm (above which an object is declared as 'flower') provides different (recall, precision) points, and a tradeoff graph between them is created. A single accuracy (or error) measurement is not useful in object detection problems since the data is highly unbalanced toward the 'negative' non-object examples, comprising above 99% of the rectangle examples in an image. Hence obtaining error of less than 1% is possible simply by always predicting a rectangle hypothesis to be negative. Instead, the area under the recall-precision curve (termed Average Precision (AP)) was used, which is the commonly used performance measurement in detection challenges.

Table 1 Correlation between the on-sight estimator of blooming intensity and human expert

Variable	Pearson correlation	Difference
$\{X_1, X_2, X_3, X_{4-8}\}$	0.82	–
$\{X_1, X_2, X_3\}$	0.81	0.01
$\{X_1, X_2\}$	0.80	0.02
$\{X_1\}$	0.78	0.04
–	0	0.82

Models with different features are depicted, trained on the 2015 dataset

X_1 is the number of flowers,, X_2 is the number of flowers squared, X_3 is the average flowers size and X_{4-8} is the number of flowers when the detection threshold is different (see blooming intensity section)

The blooming intensity estimation evaluation was done with the Pearson correlation index. The correlation between the algorithm's blooming intensity estimation and the human judgment was measured. In order to avoid over-fitting, 50 experiments were performed, wherein each experiment the dataset was randomly split into train and test sets (70% for training). The reported results of each estimator are the averaged Pearson correlation obtained across the experiments.

Results and discussion

Detection results

The obtained recall-precision curve for the flowers can be seen in Fig. 2a. The Average Precision score of this curve is 0.683, indicating a good, though not perfect, level of overall detection performance. As an example, a threshold exists which allows obtaining a detection rate of 0.695 of the flowers with average of 1.4 false alarms per sub-image. Typical examples of the algorithm's output can be seen in Fig. 2b, c.

Blooming intensity estimation

On sight estimator results

The blooming intensity of a subset containing 62 images (from 2015) was estimated by two different human experts. Table 1 shows the agreement between the on-sight estimator of the blooming intensity and the human expert, expressed as the correlation between their estimations. The most significant feature to the model was the number of flowers, with the other features providing a small but positive contribution to the Pearson correlation. The agreement between the two human experts, as also measured by the Pearson correlation between them, and was found to be 0.8, indicating a high but not perfect degree of agreement. The correlation between the on-sight algorithm estimation and the human estimation was 0.93 and 0.82 for an experienced and a less trained expert respectively. This means that on this dataset the algorithm has higher agreement with the two humans than they have among themselves—hence it qualifies as an expert. Note also that the agreement between the algorithm and a mature expert was higher than the agreement with a less trained expert.

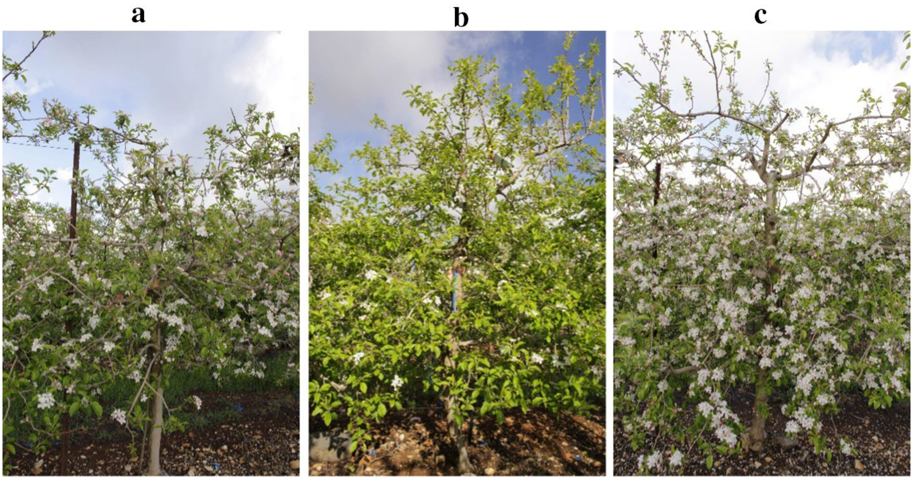

Fig. 3 Human judges and algorithm agreement and disagreement of blooming intensity from 2015. **a**, **b** trees on which both experts and the algorithm agreed on the blooming intensity (human judges ranked tree A as 3–3.5 and tree B as 1–1.5 while the algorithm ranked them 3.5 and 1.5 respectively. **c** the human judges agreed on the blooming intensity and ranked it to be 4.5–5, the algorithm ranked the blooming intensity as 3.5

Table 2 Results of the sequenced-based estimator, based on the 2015 dataset, annotated by a less trained expert

Variable	Pearson correlation	Difference
$\{X_1, X_2, X_3, X_{4-8}, X_9, X_{10}, X_{11}, X_{12}\}$	0.86	–
$\{X_1, X_2, X_3, X_{4-8}, X_9, X_{10}, X_{11}\}$	0.85	0.01
$\{X_1, X_2, X_3, X_{4-8}, X_9, X_{10}\}$	0.84	0.01
$\{X_1, X_2, X_3, X_{4-8}, X_9\}$	0.82	0.04

The description of X_1, X_2, X_3, X_{4-8} can be found in Table 1, with X_1 being the dominant feature of number of flowers. X_9 is the day index in which the image was taken, X_{10} is the difference between the numbers of flowers in last two consecutive days, X_{11} is the difference between the numbers of flowers in last two consecutive days squared and X_{12} is the difference between the current number of flowers and the average number of flowers in previous days squared

In Fig. 3, examples for agreement and disagreement between the algorithm and the human judges are presented. The on sight estimator correlation on the whole 2014 dataset (300 images) was 0.88, and 0.82 for the 2015 dataset (795 images). Table 1 shows the Pearson correlation agreement scores of model versions for 2015.

Sequence-based estimator results

Since the apple trees blooming season was 10 days long, the use of a sequence-based estimator can help estimate the blooming intensity in case there are 'hidden' features related to the time series structure. Table 2 shows the correlation between the sequenced-based estimator and a less trained expert for the 2015 dataset. The results show that time series

Table 3 Pearson correlation obtained by using the *on-sight* estimator when trying to transfer the knowledge between years

		Test-on	
		2014	2015
Train-on	2014	0.88	0.81
	2015	0.88	0.82

Table 4 Pearson correlation obtained by using the *sequenced-based* estimator when trying to transfer the knowledge between years

		Test-on	
		2014	2015
Train-on	2014	0.88	0.80
	2015	0.83	0.86

features like the day index (X_9), and the difference between the numbers of flowers in last two consecutive days (X_{10}) significantly contribute to the prediction. Using the sequenced-based estimator, the Pearson correlation obtained was 0.88 in 2014 data and 0.86 in 2015. The features used in the sequenced-based model are described in Table 2. This indicates the contribution of time-series factors to the bloom intensity estimation as done by humans, and a relative success of the algorithm in mimicking this logic.

Knowledge transfer between years

An important concern for model generality was the ability of a trained model (which was trained on a specific year dataset), to predict the blooming intensity of a different year. Such transfer is necessary for a realistic blooming estimation system, which does not require re-training every year. Tables 3 and 4 shows the Pearson correlation obtained, using the on-sight and the sequenced-based estimator, both with all features, when cross-year generalization is considered between 2014 and 2015. For example, the sequence-based model trained on the 2015 dataset resulted a correlation of 0.83 on the 2014 dataset, compared to 0.88 obtained by full training and testing on 2014 data. The small gap between the results indicates that the model is, to a large degree, year-invariant. It can be seen that both the on-sight and the sequence-based estimator generalized well between years, but the on-sight model, which is simpler, had an advantage in this respect.

Comparison against baselines approachs

The proposed model in this work was compared to the color-based baselines described in the 'methods compared' section. The threshold for declaring a pixel as 'flower-pixel' was tuned to maximize the Pearson correlation of the resulting regressor (with the human judgment). The results are described in Table 5. It can be seen that the Euler-based system is superior to plain pixel counting. However, both pixel based and Euler-number based methods perform lower than the CNN-based system, with their best results reaching 0.45 Pearson coefficient, while the CNN-based system reached above 0.8 in all cases. In addition, the color-based system suffered from instability across the 2 years tested, with the HSV–based systems providing better results for 2014, while for 2015 RGB was superior.

Table 5 Comparison between Pearson correlation scores of color based methods and the proposed CNN-based models (on-sight and sequenced-based)

	RGB		HSV	
	2014	2015	2014	2015
Pixel based	0.11	0.39	0.32	0.19
Euler number based	0.17	0.45	0.43	0.19
On-sight	0.88	0.88	–	–
Sequenced-based	0.82	0.86	–	–

The CNN-based models provide agreement rates (with the human expert) which are in average more than twice better than color-based methods

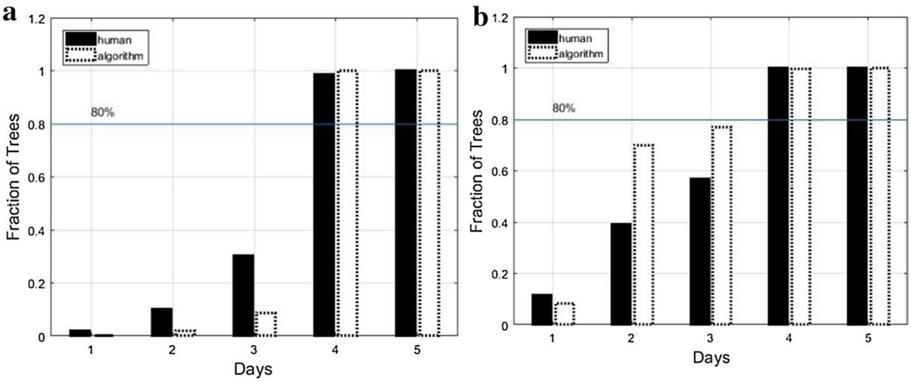

Fig. 4 Comparison of peak determination between human annotator and the sequenced-based estimator in 2014 (**a**) and in 2015 (**b**). The graphs present a CDF of the number of trees reached to their peak in each day. The solid black bars represent the human estimation and the dotted bars the algorithm judgment

Blooming peak date estimator

The global orchard's blooming peak date (as determined by the human annotator) and the sequenced-based algorithm estimation are presented in Fig. 4a, b respectively. In both years the algorithm was able to successfully determine the orchard's day of peak blooming, defined as the day in which 80% of the trees reached their peak. The day found by the peak finding algorithm (for both algorithmic estimations and human annotations) was indeed the day determined the breeders as the orchard's peak date.

This work strives to go beyond global peak date decision and tries to assess if a per-tree peak date estimation is feasible. Though both estimators agreed with the human annotator regarding the global peak date, agreement on blooming peak for individual trees was lower. Figure 5 show histograms of the deviations in peak date determination between days inferred from the human annotations and days inferred from the on-sight blooming intensity estimator's scores. Table 6 summarizes the same results as "Hit Ratio", which is defined the percentage of trees in which the algorithm and the human judge agree on the peak blooming day of the tree, and "Hit Plus/Minus 1", which represent the percentage of trees in which the judgments differ by one day at most. The table contains results for two choices of the peak determination threshold ε (see the blooming peak estimator section). For $\varepsilon = 0.5$, the human and the on-sight estimator fully agreed

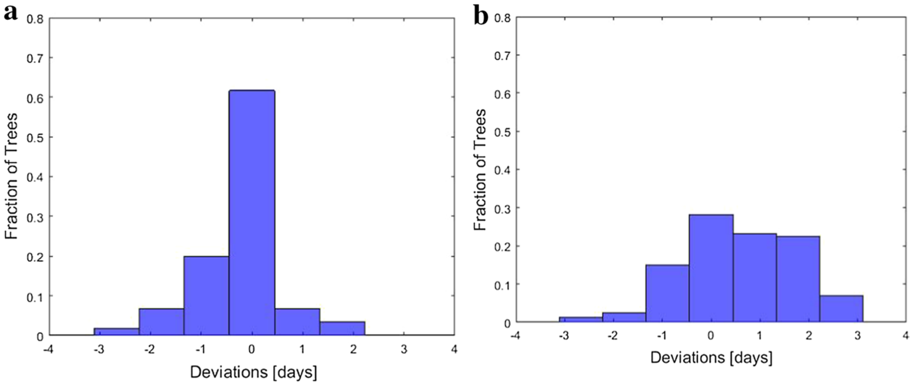

Fig. 5 Deviation of single tree peak estimation when using the on-sight estimator. The difference threshold (see the blooming peak estimator section for more details) was set to 0.5 for the on-sight estimator and 0 for the human annotator. The agreement between the human annotator and the estimator on most of the dataset was ~ 60% in 2014 (**a**), but in 2015 (**b**), the agreement was lower

Table 6 Agreement on specific tree peak date in two cases

		% Hit		% Hit	
		2014	2015	2014	2015
Peak Sensitivity	$\epsilon = 0.5$	0.57	0.33	0.92	0.69
	$\epsilon = 0.75$	0.47	0.58	0.93	0.84

Row 1: The difference threshold was set to 0.5 for the on-sight estimator and 0 for the human annotator. Row 2: The difference threshold was set to 0.75 for the on-sight estimator and 0.5 for the human annotator

on the peak date in 47–57% of the trees, and for ~ 90% of the trees deviations were one day at most.

Discussion

The blooming intensity estimators presented in this work tried to follow human decision making and considerations. Though the blooming intensity estimation of the algorithm reached human level performance for the on-sight estimator, there are still things that can be improved:

1. Improving the detection and counting of open flowers—selected branches with accurate human counts of open flowers could be followed over time to increase the system accuracy.
2. Detection and counting of buds—the buds on the tree provide information about its maturity. Abundance of them indicates that the tree is in its first blooming stages, while their absence indicates the contrary. Also, buds detection may enable prediction of the blooming peak day a few days before peak onset, which may be very useful for work planning.

3. Isolation of the trees in the images—the images were taken in real outdoor settings, where sometimes branches of neighboring trees appear in a tree image. Better tree isolation can be enforced using either a better image capture protocol or development of tree segmentation algorithms.

A main obstacle for further improvement of the on-sight estimator is that it is already in a human level, and further improvements will be very hard to estimate by comparing to human judgment, which already seems to be noisier than the algorithm. Hence further improvement may require a more objective methodology for algorithm assessment, for example, exact counting of flowers for a tree population. Going beyond blossom intensity estimation, a good option is an experiment in which the real number of apples obtained from a tree is measured, for different thinning decisions suggested by human and algorithm.

One complex issue is the time-series logic applied by the human expert in its blooming intensity estimation. This logic is not fully understood, and it's hard to tell how replicable and reliable it is. The latter point can be estimated in future studies by collecting annotations from several experts on the same sequence dataset, and see if they agree with each other better than the agreement between algorithm and human. This will help to clarify if the time series human considerations are replicable and consistent. If this logic is consistent, a possible direction for further research would be improvement of the algorithmic time series reasoning, so it has higher resemblance to the human reasoning. However, in this case too, it is not clear if the human sequential considerations, even if replicable, are indeed relevant for better thinning decisions.

This work showed that blooming intensity and peak-blooming date can be determined algorithmically, with close to human performance. This can be taken into practice by developing a real system, combining the suggested perception system with a sprayer equipped with an adaptable spray device. Toward this goal, however, additional perception challenges should be met, like spraying uniformity, self-navigating systems and tree isolation.

Summary and conclusions

The proposed system consists of three modules—flower detection and counting, blooming intensity estimation, and blooming peak estimation. Based on the results presented, a CNN-based detector was able to detect flowers reliably despite confounding conditions including flower viewpoint, illumination variance, flower clustering formation, and flower-occlusion. A CNN-based visual inference system can get close to human accuracy in the task of blooming level estimation. Furthermore, using only 15 large scale trees images for training, the obtained performance was comparable to other detection benchmarks datasets, containing a larger amount of images. In addition, a system built for flower level estimation in a certain year was robust enough to provide good results in another year. Finally, an inference mechanism with such a visual system can obtain close to human performance in the task of choosing the blooming peak date.

Blooming intensity estimation is a statistical task: its success does not depend on accurate detection of all flowers, but on rough estimation of their number. It is hence not sensitive to errors typical for current object detection technology, and its accuracy is enough to reach human level performance, as reported and shown for the on-sight estimator. However, when time series effects also exist there are still discrepancies between the algorithm (even its sequential version) and the human judgment. The human sequential judgment is not explicit, hard to justify and it is not clear if it is repeatable, i.e. if the same expert annotates the same dataset twice, will he give the same estimation.

Regarding the task of determining blooming peak date, though the global peak date for the entire orchard was found, the algorithm has certain disagreements with the human judgment regarding the blooming peak date of individual trees. These disagreements, however, are of a single day in most cases (> 90%). This finding may pave the way toward tree-specific thinning, especially if the whole process can be automated with mobile vehicles which close the loop and enable thinning of different trees in different days without extra human effort. More research is required to understand if the current algorithm's accuracy is enough for tree-specific thinning and to estimate the possible benefits of such policy.

References

Adamsen, F. J., Coffelt, T. A., Nelson, J. M., Barnes, E. M., & Rice, R. C. (2000). Method for using images from a color digital camera to estimate flower number. *Crop Science, 40*(3), 704–709.

Aggelopoulou, A. D., Bochtis, D., Fountas, S., Swain, K. C., Gemtos, T. A., & Nanos, G. D. (2011). Yield prediction in apple orchards based on image processing. *Precision Agriculture, 12*(3), 448–456.

Bargoti, S., & Underwood, J. (2017, May). Deep fruit detection in orchards. In *IEEE International Conference on Robotics and Automation (ICRA), 2017* (pp. 3626–3633). IEEE.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009.* (pp. 248–255). IEEE.

Dennis, F. J. (2000). The history of fruit thinning. *Plant Growth Regulation, 31*(1–2), 1–16.

Dobrescu, A., Giuffrida, M. V., & Tsaftaris, S. A. (2017). Leveraging multiple datasets for deep leaf counting. *arXiv preprint* arXiv:1709.01472.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*(2), 303–338.

Forshey, C. (1986). Chemical fruit thinning of apples. New York's food and life science bulletin, bulletin number 116. Published by New York State Agricultural Experiment Station. Retrieved from https://ecommons.cornell.edu/bitstream/handle/1813/5147/FLS-116.pdf?sequence=1&isAllowed=y. Last Accessed June 2019.

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2015). Sensors and systems for fruit detection and localization: A review. *Computers and Electronics in Agriculture, 116,* 8–19.

Gourley, J. H. (1922). *Textbook of Pomology*. New-York: Macmillan.

Hočevar, M., Širok, B., Godeša, T., & Stopar, M. (2014). Flowering estimation in apple orchards by image analysis. *Precision Agriculture, 15*(4), 466–478.

Hosang, J., Benenson, R., & Schiele, B. (2017). Learning non-maximum suppression. *arXiv preprint* arXiv:1705.02950.

Kapach, K., Barnea, E., Mairon, R., Edan, Y., & Ben-Shahar, O. (2012). Computer vision for fruit harvesting robots–state of the art and challenges ahead. *International Journal of Computational Vision and Robotics, 3*(1–2), 4–34.

Koen, B. V. (1985). *Definition of the Engineering Method*. Washington, DC: ASEE Publications.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems, 25,* 1097–1105.

Link, H. (2000). Significance of flower and fruit thinning on fruit quality. *Plant Growth Regulation, 31*(1–2), 17–26.

Linker, R., Cohen, O., & Naor, A. (2012). Determination of the number of green apples in RGB images recorded in orchards. *Computers and Electronics in Agriculture, 81,* 45–57.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440).

Minervini, M., Fischbach, A., Scharr, H., & Tsaftaris, S. A. (2016). Finely-grained annotated datasets for image-based plant phenotyping. *Pattern Recognition Letters, 81,* 80–89.

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science, 7,* 1419.

Potena, C., Nardi, D., & Pretto, A. (2016, July). Fast and accurate crop and weed identification with summarized train sets for precision agriculture. In *International Conference on Intelligent Autonomous Systems* (pp. 105–121).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91–99).

Robinson, T. L., Lakso, A. N., & Hoying, S. A. (2010, August). Advances in predicting chemical thinner response of apple using a MaluSim carbon balance model. In *XXVIII International Horticultural Congress on Science and Horticulture for People (IHC2010): International Symposium on Plant 932* (pp. 223–229).

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). Deepfruits: A fruit detection system using deep neural networks. *Sensors, 16*(8), 1222.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint* arXiv:1409.1556.

Wang, Q., Nuske, S., Bergerman, M., & Singh, S. (2013). Automated crop yield estimation for apple orchards. In *Experimental robotics* (pp. 745–758).

Yoder, K., Yuan, R., Combs, L., Byers, R., McFerson, J., & Schmidt, T. (2009). Effects of temperature and the combination of liquid lime sulfur and fish oil on pollen germination, pollen tube growth, and fruit set in apples. *HortScience, 44*(5), 1277–1283.