

Robust node detection and tracking in fruit-vegetable crops using deep learning and multi-view imaging

Biosystems Engineering

Boogaard, Frans P.; Rongen, Kamiel S.A.H.; Kootstra, Gert W.

<https://doi.org/10.1016/j.biosystemseng.2020.01.023>

This article is made publicly available in the institutional repository of Wageningen University and Research, under the terms of article 25fa of the Dutch Copyright Act, also known as the Amendment Taverne. This has been done with explicit consent by the author.

Article 25fa states that the author of a short scientific work funded either wholly or partially by Dutch public funds is entitled to make that work publicly available for no consideration following a reasonable period of time after the work was first published, provided that clear reference is made to the source of the first publication of the work.

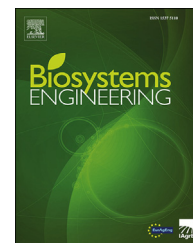
This publication is distributed under The Association of Universities in the Netherlands (VSNU) 'Article 25fa implementation' project. In this project research outputs of researchers employed by Dutch Universities that comply with the legal requirements of Article 25fa of the Dutch Copyright Act are distributed online and free of cost or other barriers in institutional repositories. Research outputs are distributed six months after their first online publication in the original published version and with proper attribution to the source of the original publication.

You are permitted to download and use the publication for personal purposes. All rights remain with the author(s) and / or copyright owner(s) of this work. Any use of the publication or parts of it other than authorised under article 25fa of the Dutch Copyright act is prohibited. Wageningen University & Research and the author(s) of this publication shall not be held responsible or liable for any damages resulting from your (re)use of this publication.

For questions regarding the public availability of this article please contact openscience.library@wur.nl

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/issn/15375110

Research Paper

Robust node detection and tracking in fruit-vegetable crops using deep learning and multi-view imaging

Frans P. Boogaard^{1,*}, Kamiel S.A.H. Rongen¹, Gert W. Kootstra

Wageningen University & Research, Farm Technology Group, P.O. Box 16, 6700 AA, Wageningen, the Netherlands

ARTICLE INFO

Article history:

Received 23 July 2019

Received in revised form

17 January 2020

Accepted 30 January 2020

Keywords:

Digital plant phenotyping

Internode length

Tracking node development

Deep learning

Multi-view imaging

Obtaining high-quality phenotypic data that can be used to study the relationship between genotype, phenotype and environment is still labour-intensive. Digital plant phenotyping can assist in collecting these data by replacing human vision by computer vision. However, for complex traits, such as plant architecture, robust and generic digital phenotyping methods have not yet been developed. This study focusses on internode length in cucumber plants. A method for estimating internode length and internode development over time is proposed. The proposed method firstly applies a robust node-detection algorithm based on a deep convolutional neural network. In tests, the algorithm had a precision of 0.95 and a recall of 0.92. The nodes are detected in images from multiple viewpoints around the plant in order to deal with the complex and cluttered plant environment and to solve the occlusion of nodes by other plant parts. The nodes detected in the multiple viewpoint images are then clustered using affinity propagation. The predicted clusters had a homogeneity of 0.98 and a completeness of 0.99. Finally, a linear function is fitted, which allows to study internode development over time. The presented method was able to measure internode length in cucumber plants with a higher accuracy and a larger temporal resolution than other methods proposed in literature and without the time investment needed to obtain the measurements manually. The relative error of our complete method was 5.8%. The proposed method provides many opportunities for robust phenotyping of fruit-vegetable crops grown under greenhouse conditions.

© 2020 IAGrE. Published by Elsevier Ltd. All rights reserved.

1. Introduction

Plant scientists and breeders study the interactions between the genotype, phenotype and environment to improve crop performance with respect to, for instance, yield, resistances to

(a)biotic stress and resource-use efficiency. Understanding this relationship allows to select plants or crossings based on their genetic potential instead of their current phenotypic expression (Houle, Govindaraju, & Omholt, 2010). Modern breeding methods apply this knowledge to increase the genetic gain of the breeding process and accelerate the

* Corresponding author.

E-mail address: frans.boogaard@wur.nl (F.P. Boogaard).¹ These authors contributed equally to this paper.<https://doi.org/10.1016/j.biosystemseng.2020.01.023>

1537-5110/© 2020 IAGrE. Published by Elsevier Ltd. All rights reserved.

Nomenclature

\cap	Intersection
\cup	Union
2D	Two dimensional
3D	Three dimensional
A	Numbering accuracy [-]
BP	Bounding box p
c	Completeness [-]
DL	Deep learning
e_i	Error between estimated internode length and ground-truth internode length [mm]
$e_{ij, \text{ accurate}}$	Error of the accurate manual method [mm]
$e_{ij, \text{ rough}}$	Error of the rough manual method [mm]
FN	False negative
FP	False positive
h	Homogeneity [-]
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
IOU	Intersection over union [-]
NP	Total number of node predictions
NP^c	Number of correctly numbered node predictions
q_x	Horizontal scaling factor for affinity propagation [-]
q_y	Vertical scaling factor for affinity propagation [-]
r_i	Relative error between estimated internode length and ground-truth internode length [%]
RGB	Red, green and blue
RMSE	Root Mean Square Error [mm]
$s_i(t)$	Ground-truth internode length between node i and node i+1 at time t [mm]
$\hat{S}_i(t)$	Estimated internode length between node i and node i+1 at time t [mm]
TP	True positive
V	V-measure [-]
x_n	Horizontal coordinate of node n [pixels]
y_n	Vertical coordinate of node n [pixels]
x'_n	Transformed horizontal coordinate of node n [pixels]
y'_n	Transformed vertical coordinate of node n [pixels]
x_n^{mm}	Metric horizontal coordinate of node n [mm]
y_n^{mm}	Metric vertical coordinate of node n [mm]
YOLO	You Only Look Once

development of improved cultivars. However, while genotypic information is now readily available due to next-generation sequencing methods, there is a lack of large amounts of accurate phenotypic data (Yol, Toker, & Uzun, 2015). The reason for this is that phenotyping is still mainly a manual process, which is time-consuming (Gehan & Kellogg, 2017) and this tends to lead to subjective data because of perception and interpretation differences among experts and experiments (Li, Zhang, & Huang, 2014). To improve this, there is a need for automated digital phenotyping methods.

Digital plant phenotyping methods mainly use computer vision to collect accurate and objective phenotypic data. The accuracy and measurement speed of these methods can match and surpass human experts (Yol et al., 2015). The most widespread methods apply two-dimensional (2D) red, green, and blue (RGB) imaging to measure morphological properties of plants. In a review by Tripodi, Massa, Venezia, and Cardi (2018), an overview of work that improves the sensor data by increasing spatial or spectral resolution of the sensors or by implementing sensors that can capture the third dimension (3D) was presented.

These developments open up possibilities to measure plant architecture on a plant organ level in fruit-vegetable crops. Plant architecture in fruit-vegetable crops consists of several traits like leaf orientation and placement, stem characteristics and fruit placement. The expression of these traits in a specific plant is the result of the genetic composition of that plant, the environmental conditions in which the plant is growing and the crop management that has been applied to the plant (Lobos et al., 2017). Understanding the influence of both genetics and environment on plant architecture (i.e. the phenotype) enables breeding towards an optimised architecture. Reasons to optimise plant architecture can for example be to increase light interception and with that plant productivity or to reduce labour, possibly by optimising the structure of the plant for automated crop maintenance or harvesting.

In this paper we focus on one of the traits defining the plant architecture, the internode length. Internode length is the distance between two consecutive nodes along the curvature of the stem. The internode length and the speed of leaf formation have a direct effect on labour requirement in the greenhouse. A uniform and stable crop development over time leads to higher efficiency in crop maintenance, which reduces the labour costs. Furthermore, variation in internode length can be an indicator of stress factors like drought and salinity (Litvin, 2009; Najla et al., 2009; Sibomana, Aguyoh, & Opiyo, 2013). By measuring internode length with a high temporal resolution, up to multiple times a day, the internode development can be tracked over time, providing detailed information about plant growth.

A current method to estimate internode length, is to divide the total plant length by the number of nodes. This is relatively fast, but it only provides a rough average per plant and does not allow to study the variation in internode lengths. Alternatively, the individual nodes can be measured using a ruler, which leads to a higher accuracy, but this increases the amount of labour required to obtain the measurements. The objective of this research is to develop an automated method for measuring internode length in cucumber that can match the accuracy of manual measurements using a ruler, but without the time investment needed to obtain the measurements manually.

Measuring in greenhouses involves certain challenges. First, due to the organisation of the greenhouse with narrow paths, a small camera-to-plant distance is required. The rows of plants furthermore cause a high amount of background clutter with both plant and non-plant material. Lighting conditions change over time due to weather conditions, artificial top and inter lighting, type of screens and glass and (self) shadowing of the plants. Because of the high occupation rate in the greenhouse,

plants also tend to occlude themselves and neighbouring plants, causing missing parts in the data (Minervini, Schar, & Tsafaris, 2015; van der Heijden et al., 2012).

Other studies on the development of automated systems to estimate internode length in fruit vegetable crops can be found in the literature. Yamamoto, Guo, and Ninomiya (2016) estimated internode length of tomato seedlings. Although plants in the seedling stage do not represent the complexity of full-grown plants, the methods used by Yamamoto et al. are relevant for this research. A multiple step 2D RGB image-processing pipeline is proposed, in which the nodes are first detected, then the order along the main stem is determined, and finally the internode length is estimated. This method is highly dependent on a pixel-wise classification algorithm that tries to find the individual nodes. Although the authors state that the images were taken in conditions close to practical cultivation, no background clutter of other plants and no occluded nodes were taken into account. In the case of full-grown plants in a high-wire growing system, these challenges cannot easily be excluded. It is therefore expected that this will lower the performance of their node-detection method in greenhouse production conditions. In this paper, the possibility of implementing a deep-learning based node-detection algorithm to improve the robustness is investigated.

Deep learning has been used in previous research in agriculture and plant phenotyping. For example, Dyrmann, Jørgensen, and Midtby (2017) proposed a method for weed detection in highly occluded cereal fields. For segmentation and counting of leaves, Giuffrida, Doerner, and Tsafaris (2018) proposed a model to count leaves from top-view plant images and Ward, Moghadam, and Hudson (2019) demonstrated that using synthetic data for training a leaf-segmentation network can outperform state-of-the-art results. Singh, Ganapathysubramanian, Sarkar, and Singh (2018) presented an overview of recent work using deep learning for phenotyping plant stress. In the work of Pound et al. (2017), the effectiveness of deep-learning methods for plant phenotyping was shown for classification of root tips and the detection of plant parts like leaf and ear tips and bases in images containing a section of a wheat shoot.

Nguyen, Slaughter, Max, Maloof, and Sinha (2015) used a structured-light multi-view method to reconstruct a 3D point-cloud model of different plants, including cucumber. The set-up consisted of multiple pairs of cameras mounted on an arc and a turn table that rotated the plant. Leaves were segmented in the point cloud using Euclidean clustering and by setting some constraints on the shape of the cluster. Internode length was estimated indirectly by the distance between two leaf centres projected on the plant's principal axis. On average, this resulted in a 7.28% error with respect to the plant height, with differences depending on the shape, size and density of the leaves. The method was tested on small plants, with a maximum height of 242 mm and was reported to work for a minimum internode length of 50 mm. The method is impractical for greenhouse conditions and the leaf-segmentation method is likely to fail often for more complex plants and cluttered conditions.

In this paper, we propose a novel method for measuring internode length and internode development over time for taller cucumber plants, up to 1.5 m. The method combines and

builds upon the ideas of Yamamoto et al. (2016) and Nguyen et al. (2015) and consists of three elements: (i) a deep neural-network for the robust detection of nodes, (ii) a combination of multiple viewpoints to deal with a complex and cluttered environment with many occlusions, and (iii) a temporal node-tracking method, which allows the study of internode development over time. The aim is to match the accuracy of the ruler-based manual measurements without the time investment required.

2. Materials & methods

In this section, the experimental set-up, the image acquisition system and the implementation of the internode length measurement method are presented.

2.1. Experimental setup

2.1.1. Plant material and reference measurements

The cucumber (*C. sativus*) variety used in this research was Proloog RZ F1 (Rijk Zwaan, De Lier, The Netherlands). Twelve plants were grown in a climate chamber in the Netherlands. According to the high-wire growing system, the plants were trained to grow vertically by placing clips to attach the stem of the plant to a supporting wire. Plants were positioned such that no occlusion between plants occurred. The measurements took place between June 25th and July 2nd in 2018. On June 25th, the plants had 7–9 nodes, which had increased to 11–13 nodes by July 2nd.

During the experiment, it was discovered that three of the twelve plants (plants four, five and eight) were not properly attached to the supporting wire and therefore bulged downwards under their own weight. Since a fixed plant-camera distance was assumed for several of the methods tested, these plants could not be properly analysed. For completeness, the results are shown with and without these outlier plants.

Ground truth data was collected by manually measuring internode length using a ruler on June 27th, June 29th and July 2nd between 07:00 H and 08:00 H. Newly formed nodes were measured once their length exceeded 10 mm. In total, 459 internode lengths were measured. This took 86 min for two persons. The average measuring time per internode length is $(86 \times 60)/459 = 11$ s. This is including time to move from plant to plant and to register the data on a laptop.

2.1.2. Camera set-up and image acquisition

Images were collected using an automated image-acquisition system that can move through the climate chamber autonomously, taking images of the cucumber plants from different viewpoints. An IMPERX B4820 16 MP CCD camera was used with an image resolution of 4904×3280 pixels (IMPERX, 2018). Relevant non-standard camera settings were the exposure time of 600 ms and the white balance mode that was set to automatic. During all image acquisition runs LED top lighting for plant growth was turned on. Two additional LED light bars were mounted, one on each side of the camera, to obtain high quality images.

The system took images from the plants from multiple viewpoints. Given a specific viewpoint, the morphological structure of the plant can result in occlusion of (part of) the nodes by leaves. Examples of visible and occluded nodes are shown in Fig. 1. Taking images from multiple viewpoints increases the chance of capturing the object of interest (in our case the node) of a plant (Hemming, Ruizendaal, Hofstee, & van Henten, 2014; Nguyen et al., 2016). Moreover, nodes can now be detected and combined in multiple images, improving the robustness of the system. Figure 2 shows a schematic overview of the camera positions used in this research. Images were taken from three camera positions (A, B and C) and six height levels (1–6). This was done from both sides of the plant gutter (0° and 180°) resulting in 36 viewpoints per plant. In Fig. 3, example images from one side of the plant gutter are shown. In section 2.3.1, the method to combine the different viewpoints is explained in more detail.

2.1.3. Image data set

Images of the 12 cucumber plants were collected using the automated image acquisition system. Measurements were done on eight consecutive days between June 25th and July 2nd at four times per day, at 05:00 H, 09:00 H, 13:00 H and 17:00 H. The total amount of images expected was $12 \text{ plants} \times 8 \text{ days} \times 4 \text{ runs} \times 36 \text{ images} = 13,824 \text{ images}$. However, due to technical issues, not all plants were photographed in all runs. Thus, in total 11,592 images were captured. All plants were imaged at least twice per day. Images taken above plant height, without visible plant material, were discarded resulting in a dataset consisting of 9990 suitable images.

2.2. Internode-length estimation

The method for estimating internode length from the collected images consists of four steps:

- 1) Detecting nodes using a deep-learning-based object-detection algorithm
- 2) Combining the detected nodes from multiple viewpoints
- 3) Clustering node detections and determining the node order
- 4) Estimating the internode length

The individual steps are explained in the following sections.

2.2.1. Step 1: detecting nodes using a deep-learning-based object-detection algorithm

Finding objects in images is a common image analysis task that has been studied for many decades. Earlier object-detection algorithms were based on hand-designed feature extractors to locate the objects of interest in the camera images. These methods usually lack generality and robustness for use in different objects and environments. With the development of deep neural networks and deep learning (DL) techniques, powerful methods have become available that allow to train object detectors in an end-to-end fashion, including the feature extraction in the learning process. This has resulted in many DL-based object-detection algorithms that have shown to be capable of detecting a wide range of objects in uncontrolled environments with varying illumination conditions, such as presented in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The advantages of DL-based approaches include being completely generic and having the ability to learn the problem from training data only (LeCun, Bengio, Hinton, & Y. LY. BG. H, 2015).

2.2.1.1. The node-detection algorithm. Among the top-performing DL-based object detection algorithms is You Only Look Once (YOLO) (Redmon & Farhadi, 2018). The most recent version, YOLO v3, outperforms the older versions for small object detection (Redmon & Farhadi, 2018) and was therefore used in this research. In addition to YOLO, many other network architectures are available. There is no clear evidence that YOLO is the best choice for our application, however, based on the comparisons carried out by Suh, Ijsselmuiden, Hofstee, and van Henten (2018) it seems reasonable to expect that different network architectures give similar results. Optimising network architecture is therefore not carried out within this research. If required in the future, the object detection algorithm can be adjusted to use an alternative network architecture relatively easily.

YOLO is a convolutional neural-network architecture that divides the input image into an $S \times S$ grid. Every grid cell is responsible for detection of objects that fall into that grid cell. For every grid cell, B bounding boxes with associated objectness and conditional class probabilities are predicted. A

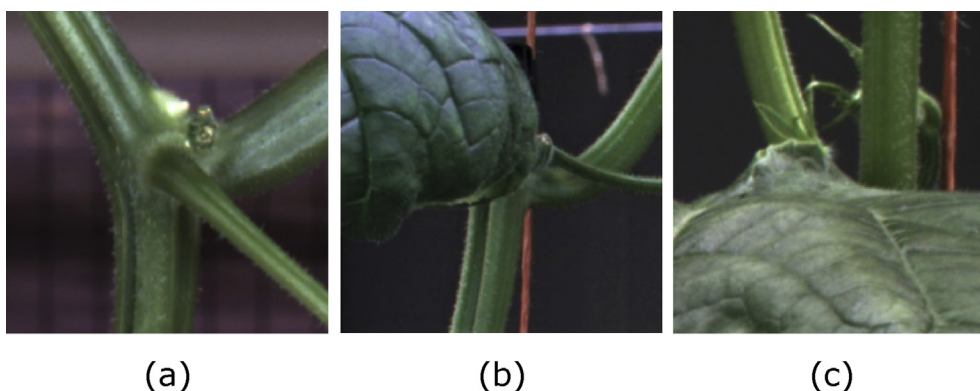


Fig. 1 – (a) completely visible node, (b) partly occluded node and (c) fully occluded node.

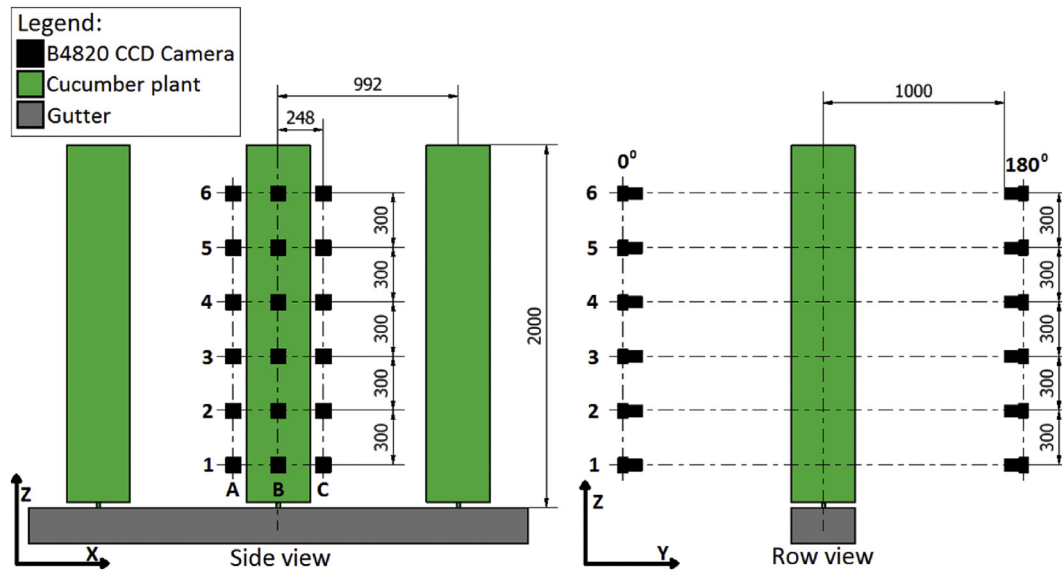


Fig. 2 – Side view and row view of the camera positions used for image acquisition. The side view image shows the camera positions A, B, and C for the 6 height levels. In the row view image it is shown that images are taken from both sides of the plant gutter. The dimensions are in mm.

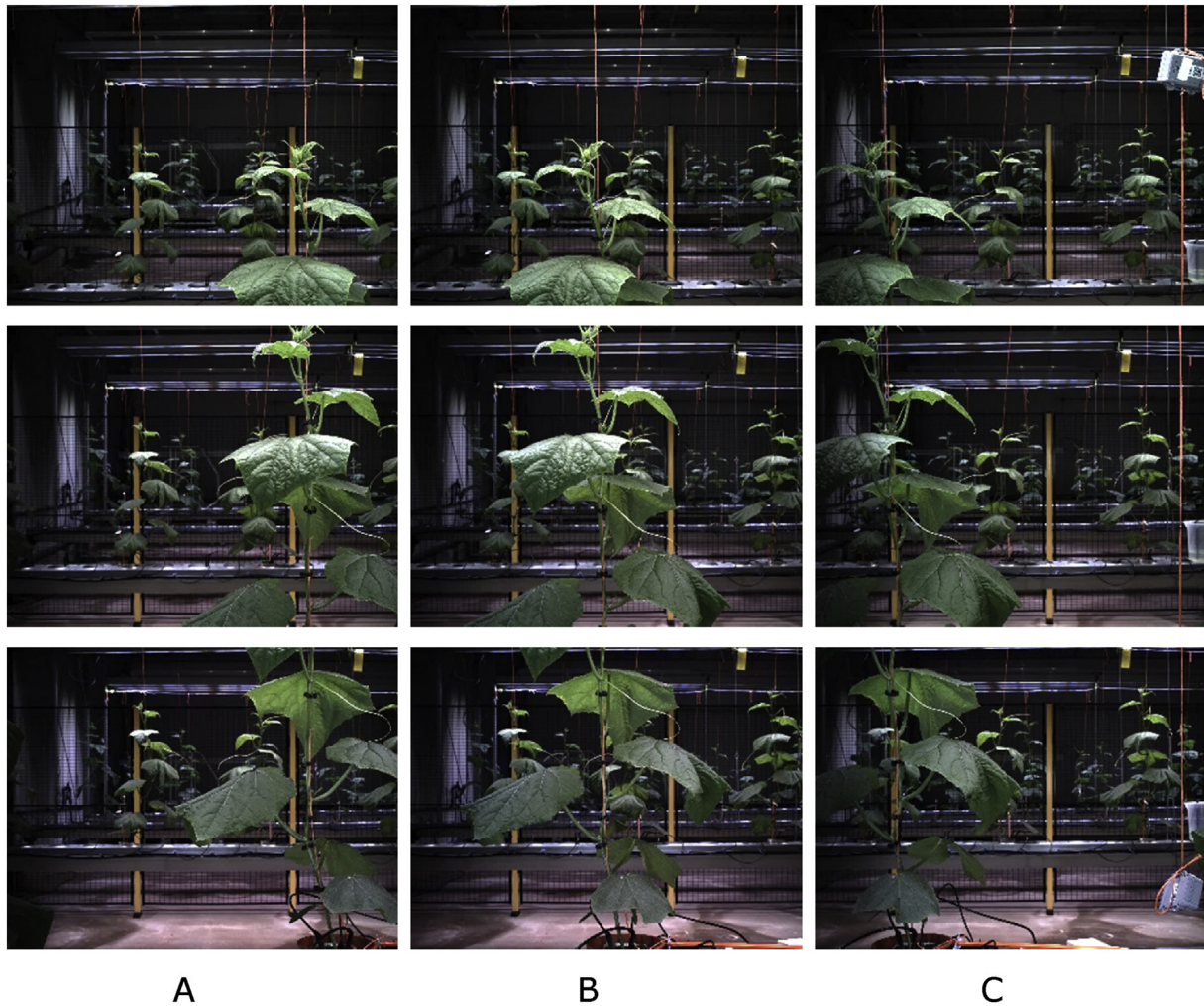


Fig. 3 – Examples of captured images from one side of the plant gutter, three camera positions and three of the six height levels.

bounding box represents the location and size of an object. Objectness is a confidence score on the probability that the bounding box contains an object and the conditional class probability gives the probability that the detected object belongs to a given class. The features are extracted from the images based on the Darknet53 (Redmon & Farhadi, 2018) architecture. The detected features are then used to predict bounding boxes at three different scales, relating to three levels of object sizes. The convolutional neural network is trained on annotated data using backpropagation (Redmon, Divvala, Girshick, & Farhadi, 2016; Redmon & Farhadi, 2017).

The YOLO v3 configuration of Alexey (2018) was used and altered concerning the pre-defined width and height to which the input images are resized (608) and the number of classes (1 class). The pre-defined width and height were set larger than the resolution used by Redmon and Farhadi (2018) to enhance performance on small objects, as nodes in our images are small objects in relation to the image size. No other adaptations of the network were done. Transfer learning was used by taking the pre-trained YOLO network that was trained on the ILSVRC dataset, using the weights file 'darknet53.conv.74' (Redmon, 2016). The pre-trained network was then further trained on images captured by our system for a maximum number of 3×10^5 training iterations. A batch size of 64 and a subdivision of 16 was used. Weights were saved every 1000 iterations.

2.2.1.2. Training of the network. In order to train the network, a training set with manually annotated nodes was generated using the Labelling annotation tool (Tzutalin, 2015). Nodes that were clearly visible were annotated as "Node", nodes that were partly occluded by plant material, the image border or any other object were annotated as "Partially occluded nodes" and fully-occluded nodes were not annotated. Only clearly visible nodes were used to train the network. For an example of the different node types see Fig. 1. Each annotated bounding box was positioned such that the centre point corresponds to the centre of the node and sized such that the Y-shaped structure of the node was included. Furthermore, the node order was included in the annotations, where node 1 is the node closest to the plant gutter and the node number increases with increasing plant height. To limit the annotation time, only a subset consisting of for each plant one randomly selected image acquisition run per day was annotated. In total 8877 fully visible nodes and 1437 partially occluded nodes were annotated.

Based on the annotated images, the network was trained on the open source neural network framework 'Darknet' (Redmon et al., 2016; Redmon & Farhadi, 2018) using an NVIDIA GeForce® GTX 1080 Ti (11 GB) GPU (NVIDIA Corporation, 2018b). CUDA version 9.0.176, cudNN version 7.1.3 (7103) and OpenCV version 2.4.9.1 were used (NVIDIA Corporation, 2018a, 2018c; OpenCV team, 2018). In order to optimally use the annotated data for training the network, a k-fold cross-validation with $k = 4$ was used. The images were split in groups of three consecutive plants as opposed to a random split of the images, to prevent that the same node would be part of both the train and validation set, which could otherwise happen as in this case one node is likely to be present on images from multiple viewpoints and different days. For all

CV-runs, the training dataset contained between 6500 and 6700 nodes and the validation dataset contained between 2100 and 2300 nodes.

2.2.1.3. Node detections. The trained network can detect multiple node instances in the images. For detection, the confidence threshold on the objectness was set to 0.5 to ensure sufficient input for the consecutive steps in the pipeline. This could have resulted in an occasional false positive, but these were filtered out by step 2 and 3, as it is very unlikely that the same false positive occurs in multiple viewpoints.

2.2.2. Step 2: combining the detected nodes from multiple viewpoints

The result of the node-detection algorithm is a list of image coordinates of the detected nodes per image (see Fig. 4a). To combine the node detections of a single plant from the different viewpoints, we need to transform the node positions into a single reference coordinate frame. For the transformation, it was assumed that all nodes lie in a plane parallel to the image planes and that the distance of the nodes to the camera positions is equal for all viewpoints. This reduces the transformation to a translation. For each camera position, the required shift in x and y direction is determined based on the shift of the manually annotated nodes. In the case of images taken from the other side of the plant gutter (180°), the images were horizontally flipped first. In this research, the offsets were determined once per plant and were used to also process images collected at later time points. The corrected position of the detected nodes is now given by the transformed coordinates (x , y), of which an example is shown in Fig. 4b.

It should be noted that the assumption of the nodes lying in a plane parallel to the camera plane at equidistance was violated by the three plants identified as "outlier" plants in section 2.1. The transformation of the node coordinates for these plants was therefore not very accurate. In future work, the transformation could be improved by considering 3D information and using calibration objects.

2.2.3. Step 3: clustering node detections and determining the node order

Because of the multiple viewpoints used in this research, a single node is likely to be detected multiple times. These detections need to be clustered and numbered before the inter-node length can be measured. Following Yamamoto et al. (2016), affinity propagation (Frey & Dueck, 2007) is used as a clustering method. Affinity propagation (AP) is a non-hierarchical clustering algorithm that automatically determines the optimal number of clusters. The algorithm requires a preference factor and a damping factor as input, which are discussed below.

Commonly, AP is used for clustering data of which both the spatial distribution and the number of clusters are unknown. However, since in our case we assume (near) vertical plant growth, some information about the spatial distribution of the nodes is available and can be used to support the clustering algorithm. Therefore, two additional factors (q_x , q_y) were introduced which were respectively multiplied by the transformed x-coordinate and the transformed y-coordinate of the detected nodes before clustering. If the value of q_x was larger

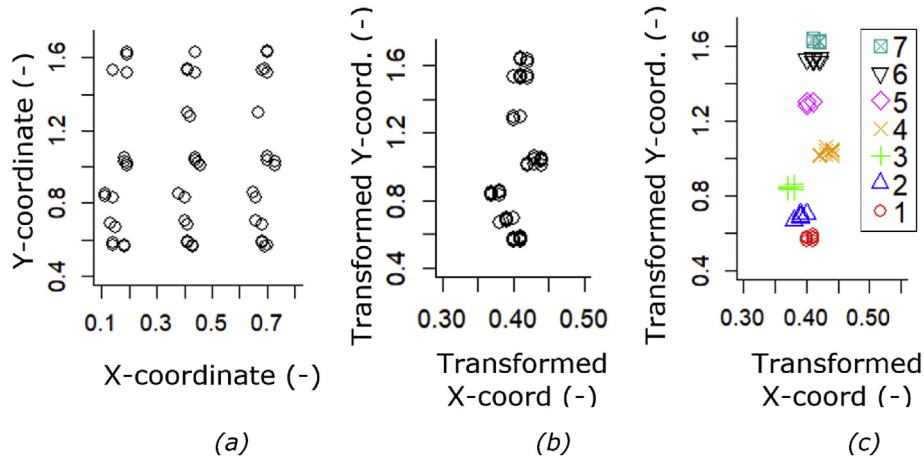


Fig. 4 – (a) Node coordinates for one plant detected in multiple viewpoints (node number unknown), (b) Detected nodes mapped onto the reference coordinate frame (node number unknown) and (c) clustered node detections numbered and in the appropriate order.

than the value of q_y , in the clustering process the variation in the horizontal (x) plane was amplified as compared to the variation in the vertical (y) plane and vice versa. Changing the values of the factors q_x and q_y allows control of the importance of differences in the x and y coordinates of the nodes. For example, if q_y is set to a high value, small differences in y-coordinate between two nodes will already cause them to be in separate clusters, while for a low value of q_y they could be clustered together.

The node-clustering algorithm was divided in three clustering steps, each with its own values for q_x and q_y , which have been determined empirically. In the first step ($q_x = 10$, $q_y = 300$), q_y was relatively high, which causes nodes with even a small difference in y to end up in separate clusters. Since it is likely that falsely detected nodes were only present in one viewpoint, these false detections will end up in clusters with only one member. The clusters with only one member were removed after the first clustering step. In the second step ($q_x = 10$, $q_y = 24$), the remaining nodes were clustered again. The values for q_x and q_y were chosen such that all detected nodes corresponding to the same physical node were clustered together. However, the difference in y-direction between detections of the youngest nodes was smaller than for the older nodes, which already had elongated internodes. The youngest nodes were therefore often clustered together in this step. To solve this, in the third step ($q_x = 10$, $q_y = 150$), the cluster with the highest y-value is re-clustered once more. If the cluster did contain two proximate nodes, they were likely to be separated in this step. In all steps, q_x was set to a low value of 10, because the plant was assumed to grow vertically and clusters should not split up horizontally (i.e. in the x direction).

To test the sensitivity of the clustering method for changes in the parameter settings, a grid-search optimisation step was carried out. The values of q_x and q_y for all three clustering steps were set at 1, 10, 100 and 500 and all possible combinations, resulting in $4^6 = 4096$ experiments. Based on the results, a second more dedicated grid search was performed in which the values for q_x and q_y in clustering step 2 were varied more densely around the empirically determined values

presented above. In this step, we tested the clustering algorithm for q_x ranging from 5 to 15 and q_y ranging from 19 to 29.

The preference factor of the algorithm controlled the number of clusters that are defined. According to the advice of Frey and Dueck (2007), this factor was set to the median of the coordinates of the bounding boxes multiplied by the values of q_x and q_y . The damping factor controls the convergence speed of the algorithm and can be used to prevent oscillations. Again according to the advice of Frey and Dueck (2007), this value was set to 0.6. Based on the results no reason was found to alter these factors.

The result of the clustering steps is a cluster of node detections for each node on the plant. Since the cluster numbering of AP is arbitrary, the clusters were re-numbered according to ascending y-coordinate of the centre of the clusters. This means that the cotyledonary node gets number 1 and counting upwards along the plant, corresponding to the order in which the nodes emerged. An example that seeks to clarify these steps is shown in Fig. 4.

2.2.4. Step 4: estimating the internode length

After clustering the detected nodes, the internode length for all time points at which images were collected could be determined. The transformed coordinates (x_i , y_i) were first converted to metric coordinates (x_i^{mm} , y_i^{mm}) using a calibration object to estimate the pixel resolution at the distance of the nodes, which in our case was $0.18 \text{ mm} \times \text{pixel}^{-1}$. The calibration object was placed in the plant gutter at the same distance from the camera as the plants. The conversion is only valid for objects at this distance and will give errors for objects at a different distance.

The estimated internode length between node i and node j at time t, $\hat{S}_{ij}(t)$ was calculated as the Euclidian distance between the centre of clusters i and j according to Eq. (1).

$$\hat{S}_{ij}(t) = \sqrt{(x_i^{mm} - x_j^{mm})^2 + (y_i^{mm} - y_j^{mm})^2} \quad [\text{mm}] \quad (1)$$

In order to obtain an estimate of the internode lengths at time points where no images were collected, a local linear function is fitted to the preceding and the subsequent measurement of a

certain time point. If a certain node was only detected once, that value was used to obtain the estimate.

2.3. Evaluation methods

The evaluation methods for the node-detection, node-clustering and cluster-numbering steps, as well as the evaluation method for the complete pipeline, are introduced in this section.

2.3.1. Multiple viewpoints

In order to evaluate the benefit of collecting images from multiple viewpoints, node detection using different combinations of viewpoints was evaluated. Eight combinations of different viewpoints (see section 2.1.2) were considered. Firstly, the effect of only imaging from one position straight in front of the plant (position B), versus imaging from three positions on one side of the plant (positions A, B and C) was analysed. Secondly, the effect of imaging from only one side of the plant (0°) or both sides (0° and 180°) was analysed. Thirdly, three different positions in the vertical direction (position 1, 3 and 5) were compared to all six vertical positions (1–6).

Two factors were evaluated: (1) the number of times a single node was observable in the different combinations of viewpoints introduced above, and (2) the percentage of nodes observable in at least two camera images, as this is the minimal cluster size used in Step 3 of our method.

2.3.2. Node detection (Step 1)

The performance of the node-detection algorithm was evaluated based on the intersection-over-union (IOU) between the predicted bounding box, B^p , and the annotated bounding box, B^a . IOU is a standard performance measure also used for the ILSVRC and COCO object detection challenges (COCO, 2018; Everingham, Van Gool, Williams, Winn, & Zisserman, 2010). The IOU value is calculated by dividing the area of the intersection of the two bounding boxes by the area of their union:

$$\text{IOU}(B^p, B^a) = \frac{|B^p \cap B^a|}{|B^p \cup B^a|} \quad [-] \quad (2)$$

IOU takes a value between 0 and 1 with higher values for better matches. The resulting IOU-value is then compared to an IOU threshold in order to decide if the node was correctly detected. A commonly used threshold of 0.5 was applied to account for some inaccuracies in the annotation (Everingham et al., 2010).

To obtain the number of true positives (TP), false positives (FP) and false negatives (FN), the IOU values between all predicted and all annotated nodes in the image were calculated and sorted from high to low. If the $\text{IOU}(B^p, B^a) > 0.5$, the prediction was a TP and the annotated node was marked as detected. The prediction was an FP if the IOU value with all annotated nodes was smaller than the IOU threshold or when the annotated node was already detected by another prediction with a higher IOU value. If an annotated node was not detected by any of the predicted bounding boxes, this was counted as an FN.

Based on the number of TP, FP and FN, the precision and recall performance measures were calculated. The precision is a measure of how many of the nodes predicted by the network correspond to an actual node, as specified in Eq. (3). The recall

describes how many of the real nodes are detected by the network, as specified in Eq. (4). The F1-score defined in Eq. (5) calculates the harmonic mean of the precision and recall, which allows to compare the performance based on a single indicator (Sasaki, 2007).

$$\text{Precision} = \frac{\# \text{ TP}}{\# \text{ TP} + \# \text{ FP}} \quad [-] \quad (3)$$

$$\text{Recall} = \frac{\# \text{ TP}}{\# \text{ TP} + \# \text{ FN}} \quad [-] \quad (4)$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad [-] \quad (5)$$

The precision, recall and F1-score were evaluated regularly during the training of the network. Evaluation of the results of the remainder of the method is based on the weights of the network obtained when the F1-score on the validation set was to its maximum.

The analysis of the results showed that the node-detection algorithm often detected nodes that were partly occluded by plant parts (e.g. see Fig. 1b) or that were partly cut off by the image border. Since annotation of these nodes was not included in the ground-truth used for training the network, these detections are strictly false positives. However, since it seems reasonable to consider the detection of partly-occluded nodes as being correct, in the evaluation of the node-detection algorithm they were classified as true positive.

2.3.3. Node clustering and cluster numbering (Step 2 and 3)

The performance of the clustering algorithm was evaluated using the Homogeneity (h), completeness (c) and V-measure (Rosenberg & Hirschberg, 2007). Homogeneity indicates to what extent each cluster contains only points of a single class and completeness indicates to what extent all points belonging to a single class are clustered in the same cluster. Both measures range from 0 (poor) to 1 (perfect). The V-measure is the harmonic mean of the homogeneity and completeness and is the performance measure used for evaluating the performance of the clustering method as specified in Eq. (6).

$$V = 2 \cdot \frac{h \cdot c}{h + c} \quad [-] \quad (6)$$

In addition, numbering accuracy was measured, as the V-measure is independent of cluster numbering. Numbering accuracy was calculated by dividing the number of correctly numbered node predictions, NP^c , by the total number of node predictions, NP , as specified in Eq. (7). The value NP^c was determined by combining the predicted node order resulting from Step 3 of our method (section 2.2.3) with the annotated node order.

$$A = \frac{NP^c}{NP} \quad [-] \quad (7)$$

2.3.4. Complete pipeline (Step 1–4)

The performance of the complete pipeline was evaluated by the error (e_i) between the estimated internode length (\hat{s}_i) and the manually measured ground-truth internode length (s_i) between node i and the consecutive node $i + 1$ (Eq. (8)). As the

image acquisition times and the manual measurements times did not coincide exactly, the estimated internode length was interpolated at the time the manual internode length measurements were performed (see Step 4, section 2.2.4). The relative error (r_i) was calculated according to Eq. (9) to enable comparison to the results of Yamamoto et al. (2016) and Nguyen et al. (2015).

$$e_i = \hat{S}_i - s_i \quad [\text{mm}] \quad (8)$$

$$r_i = \frac{|e_i|}{s_i} \quad [\%] \quad (9)$$

In order to compare the results of our method to the manual measurements, the error of the manual measurements was also estimated. This was done by comparing the manual measurement of a specific node at a specific day with the average of the manual measurements. We distinguished between the rough measurement (dividing the total plant length by the number of nodes) and the accurate measurement using a ruler. For the rough measurement, the error was estimated by taking the difference between the measured internode length and the average measured internode length per plant (taking the sum over the nodes), per day, according to Eq. (10). For the accurate measurement, the difference between the measured internode length and the average measured internode length per node per day was taken, according to Eq. (11). In both calculations, only the first five internodes were taken into account, to prevent internodes still elongating from influencing the results.

$$e_{ij, \text{ rough}} = s_{ij} - \frac{\sum_{i=1}^n \sum_{j=1}^d s_{ij}}{d \cdot n} \quad [\text{mm}] \quad (10)$$

$$e_{ij, \text{ accurate}} = s_{ij} - \frac{\sum_{j=1}^d s_{ij}}{d} \quad [\text{mm}] \quad (11)$$

where s_{ij} is the measured internode length between node i and the consecutive node $i + 1$ at day j , d is the number of days a certain internode length was measured and n is the total number of nodes taken into account (5 in our case).

3. Results

Section 3.1 presents the analysis of the number of viewpoints required for successful node detection. The results of the node-detection algorithm are presented in section 3.2. In section 3.3, the results of the node-clustering and cluster-numbering algorithms are given. The results of the full internode-length estimation method are finally presented in section 3.4.

3.1. Multiple viewpoints

The average number of observations of a node as a function of the number of viewpoints is plotted in Fig. 5. The more viewpoints taken, the more frequent the node is visible in the images acquired. When the plant is observed only from straight in front (position B) at three different heights, there are three camera images and nodes can be detected on average 0.8 times. When 36

images are taken (positions A, B and C, both sides of the plant gutter and 6 heights), the nodes can be detected on average 9.1 times.

In our algorithm, a node needs to be detected at least twice, as node clusters with only one observation were considered noise by our node-clustering method. Figure 6 shows the percentage of nodes that was identified at least twice using different viewpoint configurations. When images of the plant were taken only from straight in front (position B) at three different heights, 1.4% of the nodes was detectable. This rises to 64.0% when images were taken at six different heights. The method benefits greatly from images taken also from the rear side of the plant gutter (B both sides). Nodes are then detectable in 56.5% and 91.9% for images taken respectively at three and six different heights. When all viewpoints are taken into account (ABC both sides), the level of detectable nodes increases to 95.6% and 99.3% for respectively three and six height positions.

3.2. Node detection (Step 1)

This section describes the results of the node-detection algorithm (Step 1). The average number of iterations required in order to train the network was 20,250, corresponding to 585 epochs. After training, the average precision, recall and F1-score for the training set were 0.97 ($\sigma = 0.005$), 0.99 ($\sigma = 0.0$) and 0.98 ($\sigma = 0.004$) respectively. For the validation set, the precision, recall and F1-score were 0.91 ($\sigma = 0.01$), 0.92 ($\sigma = 0.01$) and 0.92 ($\sigma = 0.01$) respectively. As stated in section 2.3.2, these scores were corrected by considering the detections of partly visible nodes as true positives. This gave an average precision, recall and F1-score for the validation set of 0.95 ($\sigma = 0.01$), 0.92 ($\sigma = 0.01$) and 0.94 ($\sigma = 0.01$), as in Table 1.

To obtain more insight into the detections of the algorithm, Fig. 7 shows some typical examples. The first row of the image shows three correctly identified nodes. Despite the large variation in the appearance of the nodes, the network learned to detect them correctly. The second row of Fig. 7 shows three typical false positive failures of the node-detection algorithm. Image (d) shows an example where a partially occluding leaf creates image features that strongly resemble a node. This issue appears only in some viewpoints and can be dealt with easily in the subsequent multi-view steps in our algorithm. The error in image (e) was caused by a difference in size between the predicted bounding box and the ground truth bounding box, causing the IOU value to drop below the set IOU threshold of 0.5. The position of the node, however, is correct. In (f), the node was not annotated because it is only partly visible as it is cut off by the image border. The node-detection algorithm, however, did detect it correctly. This is an example of a detection that was considered as a true positive. The last row of Fig. 7 shows three false negative examples. These nodes were annotated in the ground truth data, but not detected by the algorithm.

Based on the visual evaluation of the predicted nodes, it can be seen that the node-detection algorithm had difficulties detecting the youngest node at the top of the plant. The average recall for the youngest nodes was 0.66 ($\sigma = 0.06$). This is considerably lower than the average recall obtained on all

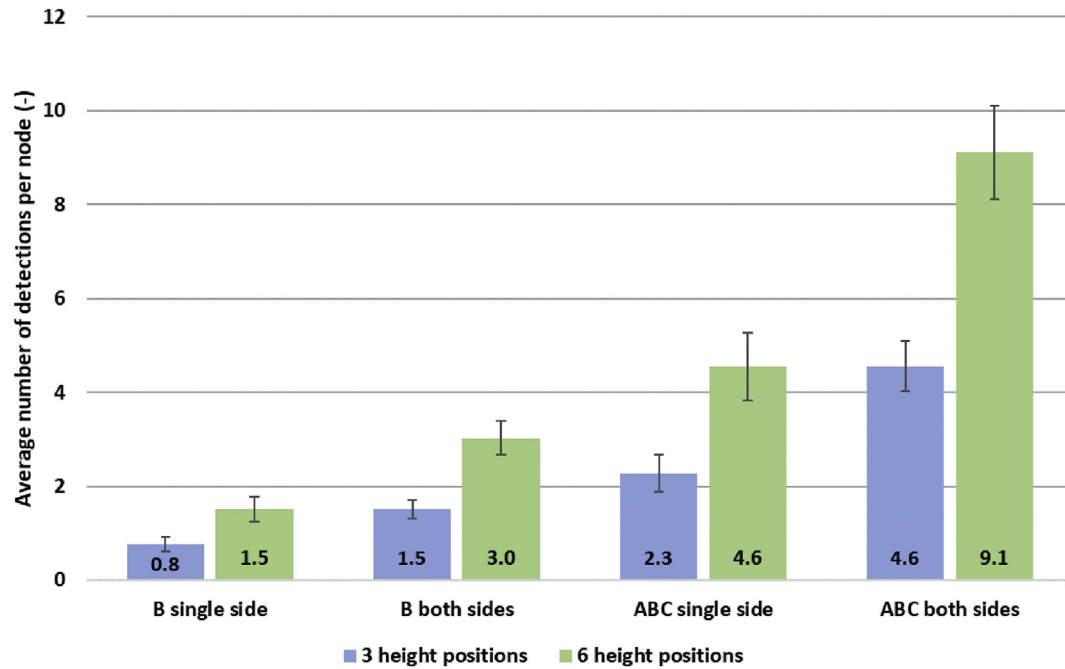


Fig. 5 – Average number of times a particular node was detected by the algorithm for different viewpoint configurations. The error bars show the standard deviation.

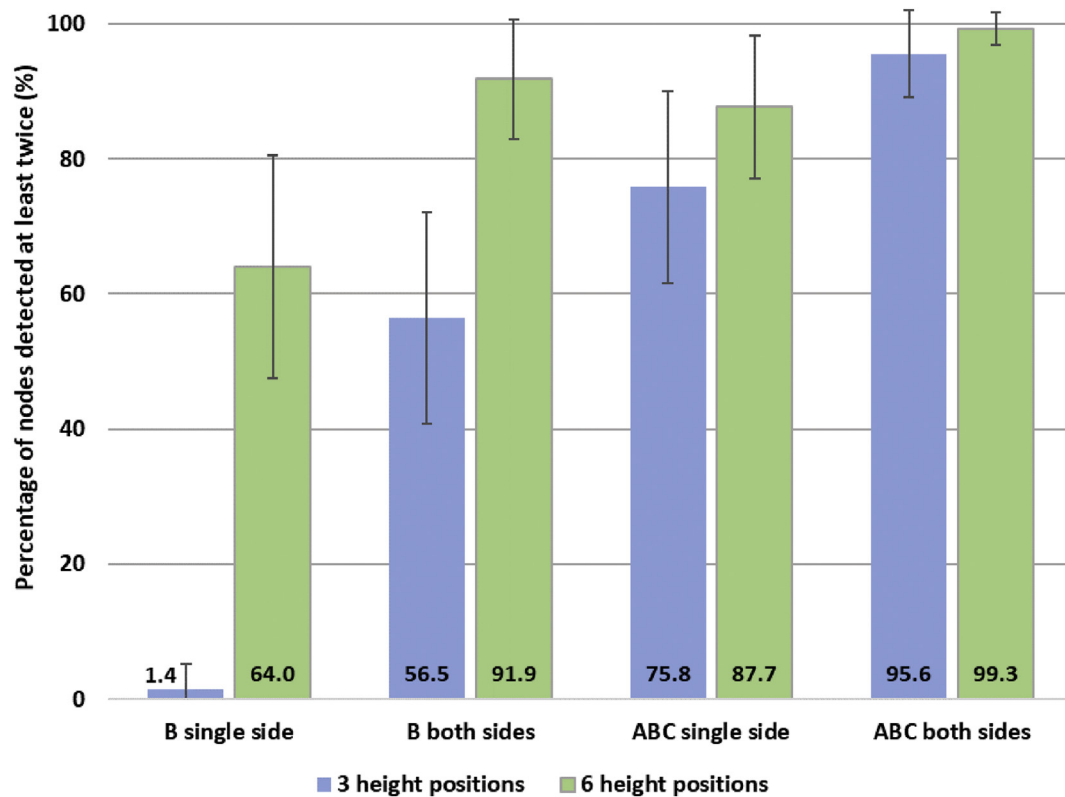


Fig. 6 – Percentage of nodes detected at least twice for different viewpoint configurations. The error bars show the standard deviation.

Table 1 – Performance measures for the trained network per cross validation run.

CV-run (–)	Validation precision (–)	Validation recall (–)	Validation F1 score (–)
1	0.93	0.91	0.92
2	0.95	0.92	0.94
3	0.96	0.93	0.94
4	0.95	0.93	0.94
Average	0.95	0.92	0.94

nodes of 0.92 ($\sigma = 0.01$), indicating that it was more difficult to find the youngest node of a plant.

3.3. Node clustering and cluster numbering (Step 2 and 3)

The node detections from the different viewpoints are combined (Step 2) and clustered by the node-clustering algorithm (Step 3). The performance of node clustering is shown in

Table. The first row shows the performance for all plants whilst the second row excludes the three outlier plants. The homogeneity and completeness are both very high, close to 1.00, illustrating that the vast majority of the clusters only represent one node and that the vast majority of detections belonging to a specific node are in the same cluster. This results in a high value for the V-measure. The results are somewhat better when the outlier plants are discarded.

Table 2 also provides the accuracy of the cluster-numbering method. The numbering accuracy for all plants is relatively low and has a high standard deviation. This is caused mainly by the outlier plants, since the numbering accuracy was close to 1.00 when the outlier plants are discarded. As explained in section 2.1.1, the outlier plants were not properly attached to the supporting wire and therefore they violated the assumption of a constant camera to plant distance. Violation of this assumption leads to incorrect node clustering and numbering. The results in Table 2 show that the clustering and numbering algorithm were able to cluster

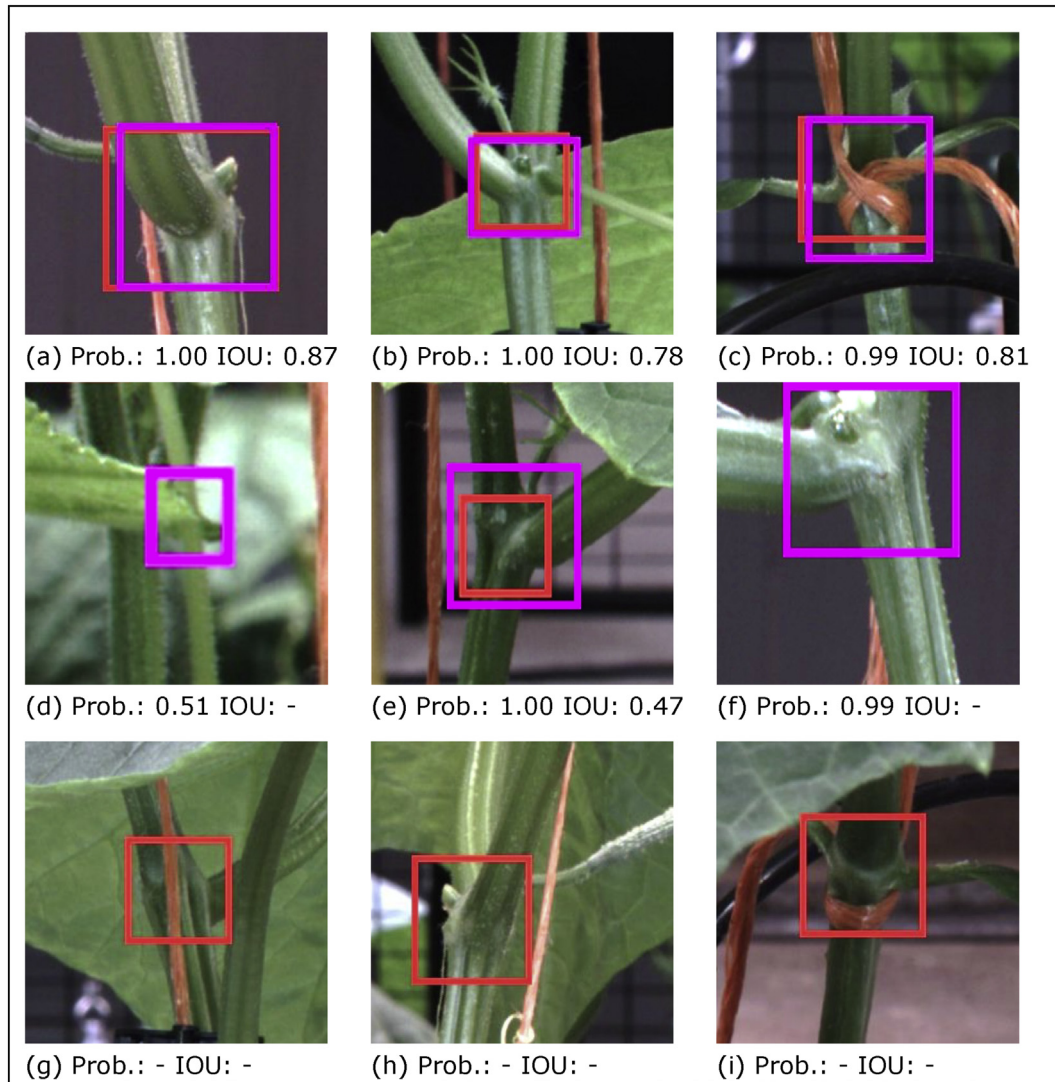


Fig. 7 – Examples of detected and undetected nodes. Purple rectangles are detections of the network and red rectangles are the ground-truth annotations. The images include true positive examples (a, b and c), false positive examples (d, e and f) and false negative examples (g, h and i).

Table 2 – Homogeneity, completeness, V-measure and numbering accuracy of the clustering algorithm.

	Node clustering						Numbering accuracy	
	Homogeneity		Completeness		V-measure		Avg.	Std.
	Avg.	Std.	Avg.	Std.	Avg.	Std.		
All plants	0.97	0.04	0.98	0.04	0.97	0.04	0.94	0.11
Excluding outliers	0.98	0.03	0.99	0.02	0.98	0.02	0.97	0.03

the annotated bounding boxes almost identical to the annotated node numbering when excluding the outlier plants from the analysis.

In Table 3, the ability of the algorithm to count the number of nodes per plant is evaluated. The total number of predicted nodes by our algorithm is compared to the actual number of nodes per plant. When all plants are taken into account, node counting was correct in 62.1% of the cases. The number of nodes was overestimated in 10.5% of the cases and underestimated in 27.4% of the cases. For all underestimations, it was found that the data points belonging to the two youngest nodes were combined into one cluster. If the outlier plants are discarded, the correct node count increases slightly to 65.3%, mainly due to the fact that an overestimation now happens in only 1.4% of the plants. Underestimation caused by the inability of the algorithm to separate the youngest nodes remained and even slightly increased. The effect on the previously reported node-clustering performance is limited, because for every youngest node there were 6–12 regular nodes.

3.3.1. Parameter check node clustering and cluster numbering

The distribution of the performance measurements obtained from the grid search, introduced in section 2.2.3, is presented in Table 4. The first step of the grid search shows that changing the values of q_x and q_y has an effect on the performance of the clustering algorithm. For all performance measurements, the maximum score found was close to 1, while the minimum score found was substantially lower. The low standard deviation shows that most values were close to the mean performance.

In the second grid search the clustering algorithm was tested for q_x ranging from 5 to 15 and q_y ranging from 19 to 29 in the second clustering step. The results are again presented in Table 4. With these settings, the standard deviation drops to below 0.0001 for all performance measurements. This shows that within the tested range the algorithm is not sensitive to small changes in the values of q_x and q_y .

The performance for a specific setting of q_x and q_y is a trade-off between the different performance evaluation criteria. In the selected settings for the remainder of our paper the empirically determined values introduced above were used, since they scored highly for all performance evaluation criteria.

3.4. Complete pipeline (Step 1–4)

3.4.1. Estimation of internode length

In Fig. 8, the internode length estimated by our algorithm (Step 1–4) is plotted against the manually measured internode length for the inlier and outlier plants. Assuming that the manual measurements are correct, in the ideal case the estimations would be equal to the manual measurements, represented by the green line. Visual assessment already shows that the estimations are closer to the ideal line for the inlier plants (left) than for the outlier plants (right). Furthermore, it seems that there is a systematic error in the data. By calibrating the data using a linear-regression fit a better model can be obtained, which is represented by the red dashed lines in Fig. 8. For the internode length estimations of the inlier plants, this results in an R^2 -value of 0.95. The Root Mean Square Error (RMSE) and the relative error (Eq. (9)) are 8.8 mm and 5.8% respectively. For the outlier plants, the R^2 -value is 0.41, while the RMSE and relative error are 24.7 mm and 32.9%. In the remainder of this paper the internode length estimations will be calibrated based on the linear regression model.

The error in the calibrated estimation of the internode length for all plants (Eq. (8)) is shown in Fig. 9. It can be seen that the distribution of the errors for the outlier plants is considerably larger than for the inlier plants. The internode length estimations for the inlier plants show a lower error and are more consistent. The medians for these plants are in the range of –2.6 mm–5.1 mm and the lower and upper quartiles range from –10.0 mm to –2.1 mm and 2.2 mm–10.5 mm respectively. For the outlier plants, the medians are in the range of –3.7 mm–12.9 mm and the lower and upper quartiles range from –36.4 mm to –8.4 mm and 8.0 mm–20.1 mm respectively.

3.4.2. Comparison to manual measurements

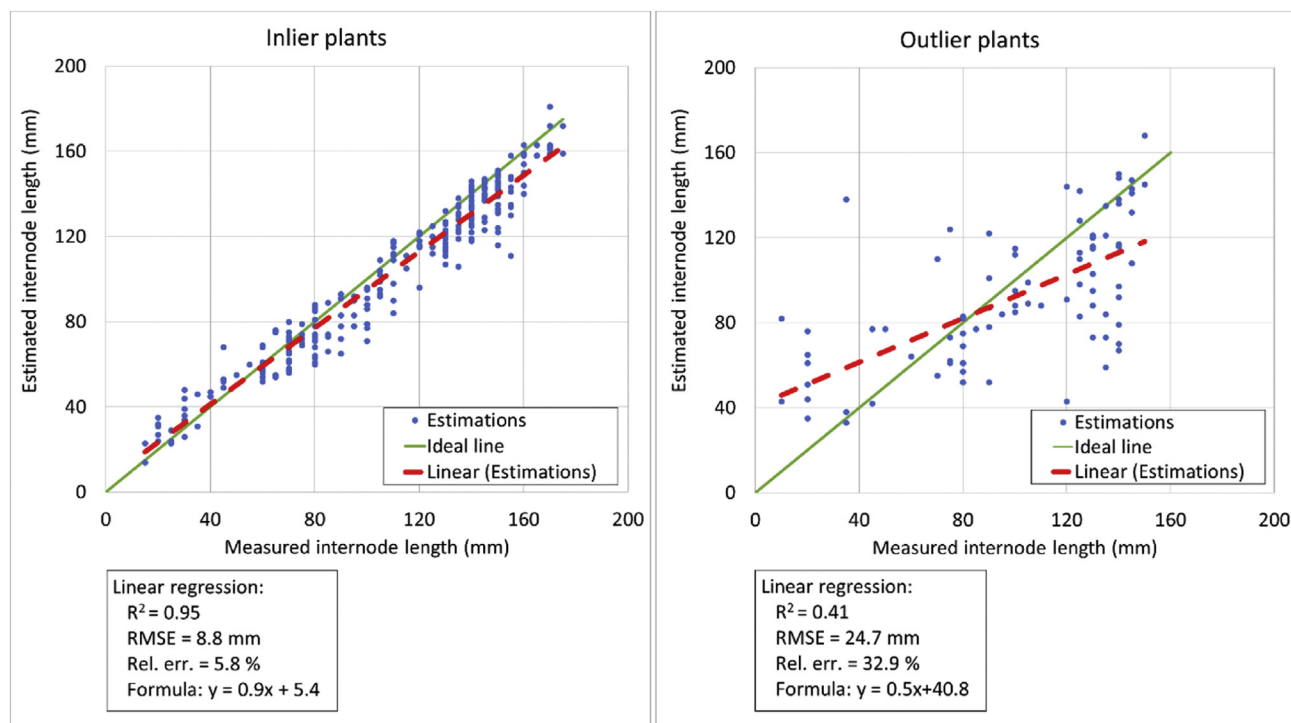
The internode length estimates made by our automatic method were compared to two methods for manual measurements, a slow and accurate method, and a fast and rough method, as explained in section 2.3.4. Figure 10a shows the errors of the manual and automatic methods. The accurate manual method showed very consistent measurements. The median error of 0 was due to the way the error was calculated according to Eq. (11). The rough manual measurement had a

Table 3 – Percentage of times that the predicted number of nodes by the clustering step is lower, equal or higher than the annotated number of nodes.

	Underestimated by 1 (%)	Equal (%)	Overestimated by 1 (%)	Overestimated by > 1 (%)
All plants	27.4	62.1	8.4	2.1
Excluding outliers	33.3	65.3	1.4	0.0

Table 4 – Results of the grid search optimisation. The numbers represent the minimum, maximum, mean and standard deviation of the performance of the clustering algorithm for different settings of q_x and q_y .

		Node clustering optimisation			Numbering accuracy
		Homogeneity	Completeness	V-measure	
Grid search step 1	Minimum	0.62	0.74	0.75	0.11
	Maximum	0.99	1.00	0.99	0.98
	Mean	0.87	0.89	0.87	0.44
	Standard deviation	0.02	0.02	0.02	0.10
Grid search step 2	Mean	0.98	0.99	0.98	0.97
	Standard deviation	<0.0001	<0.0001	<0.0001	<0.0001

**Fig. 8 – The relation between the estimated internode length and the manually measured internode length specified for inliers (plants having a (near) vertical growth pattern) and outliers.**

median error of -3.5 mm with a large spread. The lower and upper quartile are ranging from -28.0 mm to 29.0 mm. The estimates of the automatic method showed a median error of 0.1 mm and they were quite consistent, with the lower and upper quartile ranging from -6.3 mm to 6.0 mm.

The absolute errors of the different methods are shown in Fig. 10b. Whether the absolute errors of the estimates made by our algorithm were significantly different from the absolute errors of the accurate and rough manual measurement method was tested. All tests were performed in R (R Core Team, 2018). First, a Shapiro–Wilk test (Shapiro & Wilk, 1965) was performed to check the normality of the data. The p-values for the accurate and rough measurements and the estimates of our method were both less than 0.001 , indicating that it cannot be assumed that the data are normally distributed. Therefore, a Wilcoxon signed-rank test (Wilcoxon, 1946) was conducted to compare the methods. Based on the test results comparing the errors of our method with the accurate manual method ($Z = -8.1$, $p < 0.001$) and the rough manual

method ($Z = -12.1$, $p < 0.001$) the null hypothesis can be rejected in both cases. Taking into account the values plotted in Fig. 10b, it can be stated that the errors of our method were significantly lower than the errors of the rough manual method, but the errors of the accurate manual method were significantly lower than the errors of our method. In other words, our automatic method outperformed the fast and rough manual method, but not the slow and accurate manual method.

4. Discussion

This paper presents a method for automated measurements of internode length in cucumber plants. The method includes a node-detection algorithm based on deep learning, multiple viewpoints to solve occlusions in the plant and a node-clustering and node-numbering algorithm to combine detections and estimate the internode length.

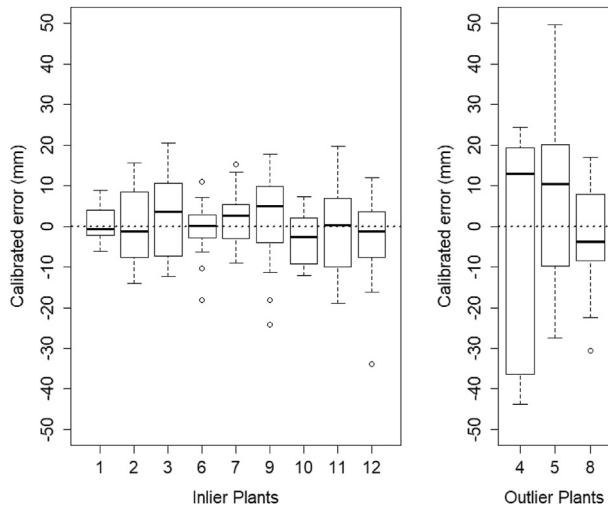


Fig. 9 – Box-and-whisker plot of the error (Eq. (8)) of the inlier (left) and the outlier (right) plants. The thick black line indicates the median, the box shows the lower and upper quartile, and the whiskers indicate the highest and lowest error, where points outside the interquartile range are considered outlier values (noted as a circle).

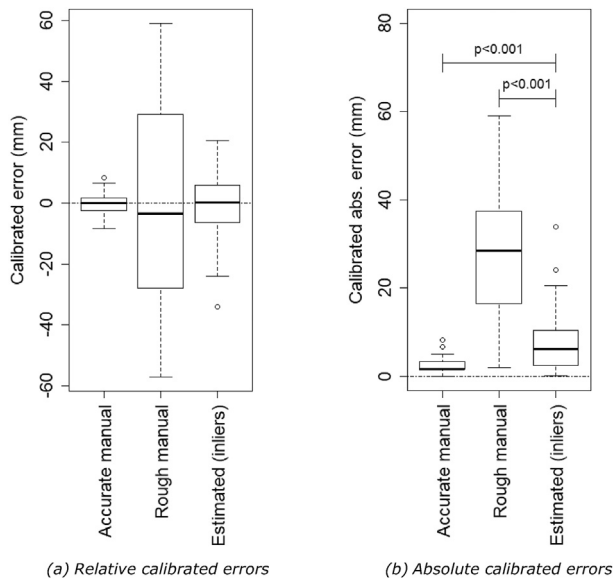


Fig. 10 – Box-and-whisker plots comparing the two manual measurement methods, accurate and rough, with the estimations from our automatic method.

The best performance was achieved by combining all different viewpoints from which images were collected. In this case, the percentage of nodes detected at least twice was 99.3%. Reducing the number of viewpoints caused the number of detected nodes to drop to as low as 1.4% for the case where images from one side of the plant and three height positions were used. Capturing images from multiple viewpoints around the plant was able to solve the problem of undetected nodes that occurs when using a single viewpoint.

The precision (0.95) and recall (0.92) of the node-detection step presented in this paper were higher than the precision (0.78) and recall (0.72) obtained by the node-detection

algorithm of Yamamoto et al. (2016), illustrating that the DL-based object-detection algorithm, applied in this research, is capable of detecting nodes more accurately than the multi-step pipeline used by Yamamoto et al. (2016). Some node detections were rejected because the size of the annotated bounding box did not match the size of the predicted bounding box (e.g. Figure 7e). Since the size of a node is difficult to define and the centre point of a node is sufficient for measuring internode length, we recommend testing point annotations instead of bounding box annotations in future work. Another interesting approach is to perform pixel-wise regression to predict heatmaps indicating areas where it is likely that a node is present. This approach was used by Pound, Atkinson, Wells, Pridmore, and French (2018) to locate and count spikes and spikelets in images of wheat.

The performance of the node-clustering and node-numbering step was heavily influenced by plants that were not properly attached to the supporting wire and therefore did not follow a near-vertical growth pattern; thus indicated as “outliers”. Current work includes testing of a system that produces 3D-data and developing a data-analysis pipeline to deal with this additional dimension. It is expected that this approach will solve the problem of underestimating the internode length by ignoring the distance between two nodes in the third dimension. In addition to an increased performance for plants having a near-vertical growth pattern, by applying this approach it is expected that also plants that had to be considered “outliers” in the current approach can be measured.

The relative error of our complete pipeline (5.8%) was lower than the relative error reported by Yamamoto et al. (2016) and Nguyen et al. (2015), which was 7.2% and 6.3% respectively. However, the calculation of this number was slightly different with these authors. In the case of Yamamoto et al. the relative error was averaged over the experimental period, which also averaged out the measurement specific error. Since our aim was to measure and follow internode development over time this is not applicable in our case. The error reported by Nguyen et al. was calculated as the error over the plant length, while our error was calculated as the error over the internode length. Recalculating our error based on the method of Nguyen et al. would not result in a fair comparison, since our plants were more than 1.6 m tall which would cause the error to drop. Therefore, it is not straightforward to compare the reported errors. Other aspects that need to be considered when comparing systems are the ability of our system to deal with occlusions and the annotation of newly formed nodes when their length exceeded 10 mm. This is shorter than the systems presented by Yamamoto et al. who focused on elongated stems of tomato, and Nguyen et al. who reported successful working algorithms where the internode length was >50 mm.

Although our method was applied to cucumber plants in this research, the only crop-specific part of the work is the training of the node-detection algorithm. With retraining of the algorithm, it is expected that our method will also perform well in other crops that are grown in similar growing systems, such as tomato or aubergine. Since similar growing systems will also result in a similar spatial distribution of the nodes, it is expected that the selected node-clustering and cluster-numbering parameters will also perform well with these crops. However, it is recommended to test this hypothesis in

future work by performing similar optimisation steps as were carried out in this research.

In line with the previous paragraph, the node-detection algorithm is the only part of the method influenced by the conditions in which the plants are growing. The plants analysed in this paper were grown in a climate chamber and were spaced out to reduce the level of occlusion. Although this is a simplification of the growing conditions in a standard greenhouse, the algorithm was able to deal with nodes that were occluded by the plant itself and was not influenced by other plants (including other nodes) and construction elements in the background of the images. However, it is expected that conditions in an actual greenhouse will still be challenging, mainly due to a higher level of variation in lighting conditions and a more complex and cluttered environment due to a higher plant density. A sufficiently large size of the training dataset should be able to solve this.

Another challenge is that in a production greenhouse the exact position of each plant is not known. Therefore, especially when measuring the same plant multiple times, a plant identification and tracking system needs to be developed. If such a system was in place, in combination with a sufficiently large training dataset, it could be expected that our approach will achieve good performance in greenhouse settings. As mentioned earlier, it is expected that implementing 3D data will improve the results both in climate chambers as well as in greenhouse settings. It is suggested to test this in future work.

5. Conclusion

Our results show the benefit of a multi-view approach. The visibility of nodes greatly improves when multiple viewpoints are used. It has been shown that the trained deep neural network was a robust method for node detection. In general, affinity propagation was able to cluster the node detections from the different viewpoints successfully. The node count in our tests was regularly underestimated due to the two youngest nodes in the top of the plant being too close to each other to be separable. However, in a later growth stage, the nodes were correctly separated. The complete method provides estimates of internode length that were more accurate than a rough manual measurement. However, there is room for improvement to achieve a performance similar to an accurate manual measurement. Including 3D-imaging is expected to improve performance greatly in future work.

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.

REFERENCES

- Alexey, A. B. (2018). Darknet. from <https://github.com/AlexeyAB/darknet/commit/cdd1cb0e8c4fda3671714bb5ad6ba1825cff16d1>. (Accessed 15 May 2018).
- COCO Consortium. (2018). COCO: Common objects in context - detection evaluation. from <http://cocodataset.org/#detection-eval>. (Accessed 30 April 2018).
- Dyrmann, M., Jørgensen, R. N., & Midtby, H. S. (2017). RoboWeedSupport - detection of weed locations in leaf occluded cereal crops using a fully convolutional neural network. *Advances in Animal Biosciences*, 8(2), 842–847. <https://doi.org/10.1017/s2040470017000206>.
- Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2), 303–338. <https://doi.org/10.1007/s11263-009-0275-4>.
- Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points (supporting online material). *Science*, 315(5814), 972–976. <https://doi.org/10.1126/science.1136800>.
- Gehan, M. A., & Kellogg, E. A. (2017). High-throughput phenotyping. *American Journal of Botany*, 104(4), 505–508. <https://doi.org/10.3732/ajb.1700044>.
- Giuffrida, M. V., Doerner, P., & Tsafaris, S. A. (2018). Pheno-deep counter: A unified and versatile deep learning architecture for leaf counting. *The Plant Journal*, 96(4), 880–890. <https://doi.org/10.1111/tpj.14064>.
- Hemming, J., Ruizendaal, J., Hofstee, J., & van Henten, E. (2014). Fruit detectability analysis for different camera positions in sweet-pepper. *Sensors*, 14(4), 6032–6044. <https://doi.org/10.3390/s140406032>.
- Houle, D., Govindaraju, D. R., & Omholt, S. (2010). Phenomics: The next challenge. *Nature Reviews Genetics*, 11(12), 855–866. <https://doi.org/10.1038/nrg2897>.
- IMPERX. (2018). IMPERX industrial cameras & imaging systems. from <https://www.imperx.com/ccd-cameras/b4820/>. (Accessed 3 July 2018).
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning review. *Nature*, 521. <https://doi.org/10.1038/nature14539>.
- Litvin, A. G. (2009). Interaction of drought stress and gibberellin metabolism on stem elongation in tomato. MSc thesis University of Georgia. from https://getd.libs.uga.edu/pdfs/litvin_alexander_g_201508_ms.pdf. (Accessed 14 August 2018).
- Li, L., Zhang, Q., & Huang, D. (2014). A review of imaging techniques for plant phenotyping. *Sensors*, 14(11), 20078–20111. <https://doi.org/10.3390/s141120078>.
- Lobos, G. A., Camargo, A. V., del Pozo, A., Araus, J. L., Ortiz, R., & Doonan, J. H. (2017). Editorial: Plant phenotyping and phenomics for plant breeding. *Frontiers of Plant Science*, 8(December), 1–3. <https://doi.org/10.3389/fpls.2017.02181>.
- Minervini, M., Scharf, H., & Tsafaris, S. A. (2015). Image analysis: The new bottleneck in plant phenotyping [applications corner]. *IEEE Signal Processing Magazine*, 32(4), 126–131. <https://doi.org/10.1109/MSP.2015.2405111>.
- Najla, S., Vercambre, G., Pagès, L., Grasselly, D., Gautier, H., & Génard, M. (2009). Tomato plant architecture as affected by salinity: Descriptive analysis and integration in a 3-D simulation model. *Botany*, 87(10), 893–904. <https://doi.org/10.1139/B09-061>.
- Nguyen, T., Slaughter, D., Max, N., Maloof, J., & Sinha, N. (2015). Structured light-based 3D reconstruction system for plants. *Sensors*, 15(8), 18587–18612. <https://doi.org/10.3390/s150818587>.
- Nguyen, T., Slaughter, D. C., Townsley, B. T., Carriedo, L., Maloof, J. N., & Sinha, N. (2016). In-field plant phenotyping using multi-view Reconstruction : An investigation in eggplant. 13th international Conference on precision agriculture.
- NVIDIA Corporation. (2018a). CUDA toolkit. from <https://developer.nvidia.com/cuda-toolkit>. (Accessed 7 May 2018).
- NVIDIA Corporation. (2018b). GEFORCE GTX 1080 Ti. from <https://www.nvidia.nl/graphics-cards/geforce/pascal/gtx-1080-ti>. (Accessed 29 May 2018).
- NVIDIA Corporation. (2018c). NVIDIA cuDNN. from <https://developer.nvidia.com/cudnn>. (Accessed 7 May 2018).

- OpenCV team. (2018). OpenCV (open source computer vision library). from <https://opencv.org/>. (Accessed 7 May 2018).
- Pound, M. P., Atkinson, J. A., Townsend, A. J., Wilson, M. H., Griffiths, M., Jackson, A. S., et al. (2017). Deep machine learning provides state-of-the-art performance in image-based plant phenotyping. *GigaScience*, 6(10), 1–10. <https://doi.org/10.1093/gigascience/gix083>.
- Pound, M. P., Atkinson, J. A., Wells, D. M., Pridmore, T. P., & French, A. P. (2018). Deep learning for multi-task plant phenotyping. In *Proceedings - 2017 IEEE international conference on computer vision workshops* (pp. 2055–2063). <https://doi.org/10.1109/ICCVW.2017.241>.
- R Core Team. (2018). R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing.
- Redmon, J. (2016). Darknet: Open source neural networks in C. Retrieved April 2, 2018, from <https://pjreddie.com/darknet/>.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only Look once: Unified, real-time object detection. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 779–788). IEEE. <https://doi.org/10.1109/CVPR.2016.91>.
- Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In *Proceedings - 30th IEEE conference on computer vision and pattern recognition* (pp. 6517–6525). <https://doi.org/10.1109/CVPR.2017.690>.
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *ArXiv*. <https://arxiv.org/abs/1804.02767>.
- Rosenberg, A., & Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 410–420). <https://doi.org/10.7916/D80V8N84>.
- Sasaki, Y. (2007). The truth of the F-measure. Retrieved from <https://www.cs.odu.edu/~mukka/cs795sum10dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf>.
- Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4), 591. <https://doi.org/10.2307/2333709>.
- Sibomana, I. C., Aguyoh, J. N., & Opiyo, A. M. (2013). Water stress affects growth and yield of container grown tomato (*Lycopersicon esculentum* Mill) plants. *Global Journal of Bio-Science and Biotechnology*, 2(4), 461–466.
- Singh, A. K., Ganapathysubramanian, B., Sarkar, S., & Singh, A. (2018). Deep learning for plant stress Phenotyping : Trends and future perspectives machine learning in plant science. *Trends in Plant Science*, 23(10), 883–898.
- Suh, H. K., IJsselmuiden, J., Hofstee, J. W., & van Henten, E. J. (2018). Transfer learning for the classification of sugar beet and volunteer potato under field conditions. *Biosystems Engineering*, 174, 50–65. <https://doi.org/10.1016/j.biosystemseng.2018.06.017>.
- Tripodi, P., Massa, D., Venezia, A., & Cardi, T. (2018). Sensing technologies for precision phenotyping in vegetable crops: Current status and future challenges. *Agronomy*, 8(4). <https://doi.org/10.3390/agronomy8040057>.
- Tzutalin. (2015). Labelling. Retrieved March 22, 2018 from <https://github.com/tzutalin/labelling>.
- van der Heijden, G., Song, Y., Horgan, G., Polder, G., Dieleman, A., Bink, M., et al. (2012). SPICY: Towards automated phenotyping of large pepper plants in the greenhouse. *Functional Plant Biology*, 39(11), 870–877. <https://doi.org/10.1071/FP12019>. Retrieved from.
- Ward, D., Moghadam, P., & Hudson, N. (2019). Deep leaf segmentation using synthetic data. In *British machine vision conference 2018* (p. 2018). BMVC.
- Wilcoxon, F. (1946). Individual comparisons of grouped data by ranking methods. *Journal of Economic Entomology*, 39(2), 269–270. <https://doi.org/10.1093/jee/39.2.269>.
- Yamamoto, K., Guo, W., & Ninomiya, S. (2016). Node detection and internode length estimation of tomato seedlings based on image analysis and machine learning. *Sensors (Switzerland)*, 16(7). <https://doi.org/10.3390/s16071044>.
- Yol, E., Toker, C., & Uzun, B. (2015). Traits for phenotyping. In J. Kumar, A. Pratap, & S. Kumar (Eds.), *The phenomics in crop plants: Trends, options and limitations* (pp. 11–26). New Delhi, India: Springer.