DIGITAL PLANT PHENOTYPING IN THREE DIMENSIONS

WHAT'S THE POINT?

Frans Pieter Boogaard

Propositions

1. Digital plant phenotyping requires more than 2 dimensions.
   (this thesis)
2. The term ‘ground truth data’ gives a false sense of the correctness of that data.
   (this thesis)
3. The popularity of publishing pre-prints shows that the review process does not function properly.
4. The ratio between rejected and accepted hypotheses in publications suggests that many results remain unshared.
5. Publication of incremental improvements of a method is pointless without specifying requirements.
6. Parenthood is a great motivator to finish a PhD trajectory.
7. Propositions without dialogue deepen the social divide.

Propositions belonging to the thesis, entitled
Digital plant phenotyping in three dimensions: what’s the point?

Frans Boogaard
Wageningen, 22 May 2023

Digital plant phenotyping in three dimensions

What’s the point?

Frans Pieter Boogaard

Thesis committee

Promotor
Prof. Dr Eldert J. van Henten
Personal chair at the Farm Technology Group
Wageningen University & Research

Co-promotor
Dr Gert W. Kootstra
Associate Professor, Farm Technology Group
Wageningen University & Research

Other members
Prof. Dr Richard Visser, Wageningen University & Research
Prof. Dr Cyrill Stachniss, University of Bonn, Germany
Prof. Dr David Rousseau, University of Angers, France
Dr Jochen Hemming, Wageningen University & Research

This research was conducted under the auspices of the Graduate School Production Ecology & Resource Conservation

Digital plant phenotyping in three dimensions

What's the point?

Frans Pieter Boogaard

Thesis
submitted in fulfilment of the requirements for the degree of doctor
at Wageningen University
by the authority of the Rector Magnificus,
Prof. Dr A.P.J. Mol,
in the presence of the
Thesis Committee appointed by the Academic Board
to be defended in public
on Monday 22 May 2023
at 4 p.m. in the Omnia Auditorium.

Frans Pieter Boogaard
Digital plant phenotyping in three dimensions: what’s the point?
190 pages

PhD thesis, Wageningen University, Wageningen, the Netherlands (2023)
With references, with summary in English

ISBN: 978-94-6447-588-3
DOI: https://doi.org/10.18174/587178

Table of contents

Page

Chapter 1 **7**
General introduction

Chapter 2 **23**
Robust node detection and tracking in fruit-vegetable crops using deep learning and multi-view imaging

Chapter 3 **57**
Boosting plant-part segmentation of cucumber plants by enriching incomplete 3D point clouds with spectral data

Chapter 4 **93**
Improved point-cloud segmentation for plant phenotyping through class-dependent sampling of training data to battle class imbalance

Chapter 5 **127**
The added value of 3D point clouds for digital plant phenotyping – a case study on internode length measurements in cucumber

Chapter 6 **155**
General Discussion, Reflection, and Recommendations

References **166**
Summary **178**
Acknowledgements **183**
PE&RC Training and education statement **187**

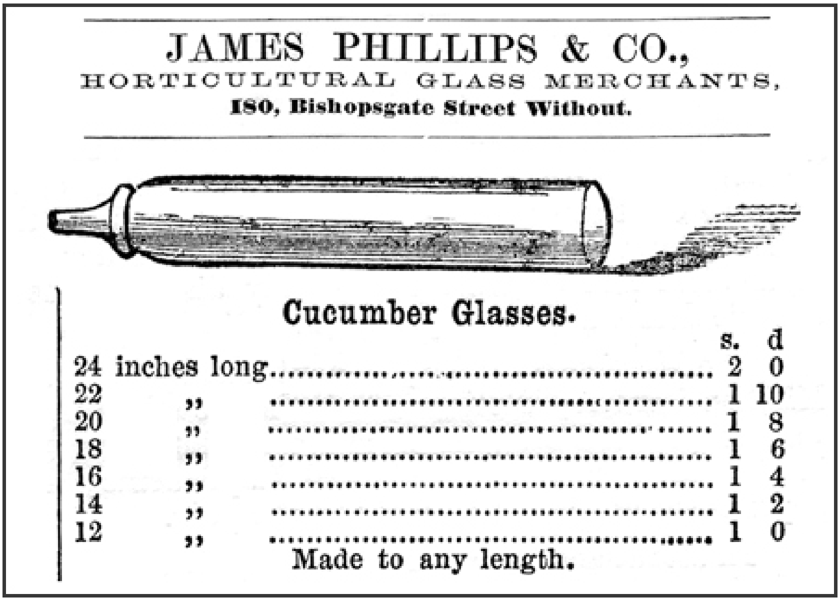
Chapter 1

General introduction

General introduction

It is said that in the 19th century, British people were not particularly fond of crooked cucumbers, as the curvy shapes of the fruits made it infeasible to cut proper slices for a sandwich. In an attempt to overcome the challenge of growing straight cucumbers, George Stephenson invented a cucumber straightener (see Figure 1.1), which was a glass sleeve that was put around a growing cucumber to force it to become "perfectly straight and level as the barrel of a gun" (Perreault, 2018).

Jumping ahead two centuries, today, plants used in horticulture are still subject to many requirements. Those requirements are set by stakeholders such as breeders, growers, retailers, and consumers and include aspects like yield, disease resistance, stress tolerance, shelf life, and taste. The expression of these traits for a certain plant is known as the phenotype of that plant. Whether the phenotype of a plant meets the requirements depends on the genetic composition of the plant and on the environment in which the plant grows. Both the environmental as well as the genetic component provide opportunities to optimise the phenotype of the plant. In the example presented above, a change in the environment, the addition of a cucumber straightener, led to a change in the phenotype: straight cucumbers.



JAMES PHILLIPS & CO.,
HORTICULTURAL GLASS MERCHANTS,
180, Bishopsgate Street Without.

Cucumber Glasses.

	s.	d
24 inches long	2	0
22 ,,	1	10
20 ,,	1	8
18 ,,	1	6
16 ,,	1	4
14 ,,	1	2
12 ,,	1	0

Made to any length.

Figure 1.1 – Advertisement for cucumber glasses to grow perfectly straight cucumbers. Image from http://www.oldgardentools.co.uk/

One of the first examples in which the genetics of the plant were changed to get a desired phenotype dates back to the early days of agriculture, approximately 10,000 years ago. Farmers selected the plants in their fields that they liked most, and used these plants to produce next seasons seeds (Wieczorek & Wright, 2012). By repeating this over and over again, desirable phenotypic characteristics that have a genetic basis were inherited to next generations. The discovery of Mendel's laws of inheritance in the 19th century started the development of genetics as a scientific field of study, enabling research to improve the process of crop improvement through plant breeding (Allard, 2019).

Outline of the general introduction

In section 1.1 of this general introduction, we first give a brief background on plant breeding. The need for high-quality phenotypic datasets is explained and the lack of these datasets is identified as the 'phenotyping bottleneck', potentially limiting the progress made in plant breeding programs. The concept of 'digital phenotyping' is then introduced as one of the technologies that could ease the phenotyping bottleneck. In section 1.2, an overview of developments in sensor technology and data science that contribute to the field of digital phenotyping is presented.

Plant architecture is the three-dimensional (3D) organisation of plant parts and was selected as a use case to explore technological developments for digital plant phenotyping. The 3D organisation of plant parts also motivates the use of 3D sensor data. The added value of using 3D data was investigated in this thesis by comparing methods based on 3D data to methods based on two-dimensional (2D) data. The use case of plant architecture is further explained in section 1.3. The digital phenotyping system to collect the 2D and 3D sensor data is then introduced in section 1.4 and the objective and main hypothesis of this thesis are defined in section 1.5. The general introduction ends with an overview of the outline of the remainder of this thesis.

1.1 Plant breeding and the phenotyping bottleneck

Plant breeding is the art and science of developing improved plant varieties (Fehr, 1991). The breeding process involves selecting and crossing plants that have specific desirable traits, to generate offspring that has a combination of those traits. Different strategies to find the optimal combination of plants to cross are available, based on the phenotype or on the genotype of the plant. In this section, we first discuss these selection strategies. The progress made in breeding programs is explained using the breeder’s equation, after which the phenotyping bottleneck is introduced as a factor limiting this progress. The concept of digital phenotyping is then presented as a technological approach to remove the phenotyping bottleneck.

Selection strategies

Phenotypic selection (PS) is a method in which plants are selected based on observable characteristics, such as size, colour, shape, or response to stress. PS is a common approach in plant breeding, especially for traits that are relatively easy to assess. However, as mentioned, the phenotype is the result of the genetic makeup as well as of the environment of the plant. So, without additional information, it is not possible to know if an observed phenotype is caused by genetic or by environmental factors. Therefore, considering that trial fields are never completely uniform, there is a risk that plants are selected that happen to be in a more suitable environment, while not necessarily having the best genetic makeup.

In contrast, marker-assisted selection (MAS) is based on the genetic composition of the plants, instead of on the observable characteristics. MAS uses one or few molecular markers as a proxy for one or few genes that have an effect on the phenotype of the plant. One of the most important aspects of MAS is unravelling the association between the markers and the phenotypic traits of interest. Finding this association depends on high quality phenotypic data and can be done through various methods such as Quantitative Trait Locus (QTL) mapping or association studies. MAS is especially useful for traits of which the desired phenotype is controlled by a single gene (monogenic traits), specifically when these traits are difficult or time-consuming to assess phenotypically.

Genomic selection (GS) is a specific form of MAS that uses a much larger amount of genomic data, mostly in the form of molecular markers. GS associates high-throughput genotyping data with phenotypic data through statistical models, to predict the genetic value of a plant for a given trait. In this way, GS allows breeders to select plants with superior genetic value for the target trait, even before the plants have been tested in the field. GS can be more accurate and efficient than PS. A breeding program based on GS is most efficient when using high quality phenotypic datasets. (Crain et al., 2018; Heffner et al., 2010; Zhu et al., 2021)

The breeder's equation

The improvement in the average value of a trait in a population, as a result of selection strategies applied in a breeding program, is known as genetic gain. The realised genetic gain can be used as a measure of the progress that is being made in a breeding program. Besides realised genetic gain, the breeder's equation can be used to estimate the expected genetic gain. The breeder's equation is often written as $R = h^2 \cdot S$, in which R is the response to selection, or the expected genetic gain, h^2 is the heritability of the trait, and S is the intensity of selection.

Heritability is a measure that indicates how much of the variation of a trait is due to genetics. A high heritability indicates that a large part of the variation in the trait is due to genetics, meaning that the trait can be improved by selection. On the other hand, a low heritability means that a large part of the variation is due to environmental factors, meaning that the expected progress due to selection is lower. The intensity of selection, S, is an indication to what degree selection for a certain trait is taking place. The selection intensity can for example be increased by selecting less plants using more stringent selection criteria for the desired trait. The selection intensity can also be increased by selecting an equal amount of plants while increasing the size of the population to select from. A higher selection intensity for a certain trait leads to a higher expected genetic gain for that trait. (Acquaah, 2007; Cobb et al., 2019; Rebetzke et al., 2019)

The phenotyping bottleneck

The phenotype of a plant is the collection of observable plant characteristics. Plant phenotyping is a scientific discipline that focuses on the assessment of these traits. Traditionally, the assessment of these traits is done by human observers. Human-based phenotyping is, besides training of the person who assesses the plants, relatively easy to implement as no machinery is required. Furthermore, the phenotypic assessment can be directly used by the observer to decide whether a plant will be selected or not.

However, the need for other phenotyping methods is increasing, as human-based phenotyping has some disadvantages. First, human-based phenotyping can be a labour-intensive task. Plant breeders typically work on a whole set of traits that need to be assessed. Furthermore, the selection intensity and thus the expected genetic gain can be increased by increasing the number of plants that is being assessed. Unfortunately, limited availability of time often leads to plant breeders focusing their phenotyping efforts on plants with the highest chance of being selected for crossings, leading to incomplete phenotypic datasets. Limited availability of time also leads to datasets with a low temporal resolution. A second issue with human-based assessments, is that the collected data tends to be subjective and sensitive to errors, due to interpretation differences between observers, or per individual observer during the course of the day. Thirdly, traits are often scored in classes having a low resolution. For example, the presence of defects on a fruit is often scored as yes or no, while a measurement of the area of each defect would provide more information. Finally, human-based assessments are limited to the human senses. For example, our eyes are only sensitive to a small part of the electromagnetic spectrum. Furthermore, accurate assessment of taste and smell requires expert panels, making the implementation on a large scale impractical.

The limitations of human-based phenotypic assessments reduce the level of genetic gain that can be achieved by plant breeders (Araus et al., 2018). As explained in the previous sections, regardless if plants are selected based on phenotype or genotype, all selection strategies would benefit from the availability of high-quality phenotypic datasets. Such datasets lead to higher heritability values, increasing the expected level of genetic gain. Furthermore, the selection intensity is limited due to the high labour requirement involved in collecting phenotypic data for large plant populations. Again, being able to collect phenotypic datasets for larger plant populations leads to a higher expected genetic gain. The current lack of the desired high-quality phenotypic datasets of large plant populations is known as the phenotyping bottleneck. (Rossi et al., 2022; Tripodi et al., 2022; Yang et al., 2020)

Digital phenotyping

Digital phenotyping uses sensors and computer algorithms to assess plant characteristics. Automation of digital phenotyping systems drastically reduces the amount of labour required to collect phenotypic data, making it possible to collect phenotypic data of larger plant populations and at a higher temporal resolution. Furthermore, depending on implementation details, computer algorithms provide objective and quantitative datasets. Plant measurements are also no longer limited to human senses. For example, cameras that are sensitive outside the visible part of the spectrum allow to extend the set of traits to beyond what is observable for humans. (Awada et al., 2018)

So, whether the focus is on reduction of labour through automated phenotyping, on higher accuracy through precision phenotyping, or on observing beyond the human senses, digital phenotyping technologies are key to increase the effectiveness and the efficiency of modern breeding programs.

1.2 Digital phenotyping to address the phenotyping bottleneck

The essence of digital plant phenotyping is to use sensors that capture data of plants and to analyse this data to extract relevant phenotypic measurements. This section is not meant to give a full overview of all digital phenotyping technology that has been presented in literature, but rather to provide some insights into the different aspects that are important to consider when working with digital phenotyping technology. For reviews of digital phenotyping technology we refer to other work. For example, the work of Tripodi et al. (2018) presents an overview focusing on phenotyping of vegetable crops in protected horticulture. In the work of Yang et al. (2020), a broader overview of progress on phenotyping is given, presenting developments that have been used in controlled environments and under field conditions, as well as for post-harvest phenotyping. Finally, the review of Yao et al. (2021) focuses on robotics for indoor and outdoor plant phenotyping. Robotics allow to automate data acquisition, which is an important aspect for increasing the scale at which a phenotyping system can be implemented, both in the size of the plant population that can be handled, as in the temporal resolution that can be achieved.

In the remainder of section 1.2, first, the focus is on what should be measured, next, sensors that can be used to collect the raw data are introduced, and finally, methods to translate the raw sensor data into plant measurements are discussed.

1.2.1 Plant phenotyping: what to measure?

Plant phenotyping can be done on different scales, from cell level, to plant parts, whole plants, or even canopies (Dhondt et al., 2013). An example of plant phenotyping on cell level is the identification and counting of cells on the epidermis of the leaf based on microscopic images (Mele & Gargiulo, 2020). On a larger scale, measurements can be obtained of detached plant parts, like leaf disks, whole leaves, or fruits, or on growing plants. These plants can be grown in environments such as climate chambers, greenhouses or open fields, each having their own advantages and disadvantages. For example, in a climate chamber, the environmental conditions can be very well managed, which is useful to study the effect of different environments on the phenotype. On the other hand, greenhouses or open fields are often more in line with commercial growing systems and it can be useful to assess the plants in these conditions also.

The most obvious parts of the plant to assess are the above-ground parts, the shoot. However, relevant traits can be identified throughout the entire lifecycle of a plant, from seed, via the shoots and the roots of the plant, to the fruit and finally, back to the seed (Yang et al., 2020). Digital plant phenotyping includes the measurement of relevant traits for all developmental stages of the plant and for all plant parts, including seed, root, shoot and fruit.

The first traits that come to mind when thinking about plant phenotyping might be visible aspects of plants, such as shape, colour and size of seeds or fruits, or the 2D or 3D architecture of the roots or the shoots of the plants. However, also aspects that are not directly visible can be assessed. For example, internal fruit quality can be assessed non-destructively with techniques such as MRI or X-ray (Nicolaï et al., 2014). Furthermore, the stress response of plants is a relevant trait to assess when breeding for more resistant or tolerant varieties. For example, a review by Tanner et al. (2022) indicates several approaches to study plant-pathogen interactions on a microscopic level, either looking at the pathogen itself or at the response of the plant to infection.

1.2.2 Sensors: how to measure?

Different types of sensors can be used to collect phenotypic data. For example, Huang et al. (2020) used sap flow sensors in a study to model evapotranspiration in cucumber and in the work of Chen and Opara (2013), several methods to asses texture in fruits and vegetables were presented. However, most digital phenotyping applications rely on visual sensors. The most frequently used sensor is a camera that captures colour images, which are analysed by computer-vision based algorithms. Three forms of resolution are relevant when working with a phenotyping system based on imaging: spatial resolution, spectral resolution, and temporal resolution.

The spatial resolution defines the ability of an imaging system to distinguish between two objects that are close to each other. In 2D sensors, the spatial resolution is determined by the size of the smallest detectable unit in both the x and y dimensions, depending on the number of pixels in the x and y dimensions. Similarly, in 3D sensors, the spatial resolution is determined by the size of the smallest detectable unit in the x, y, and z dimensions. The size of the smallest detectable unit depends on the sensor, including optics, and on the distance between the sensor and the measured object. Higher spatial sensor resolution generally means that smaller parts of the plant are identifiable in the data.

The spectral resolution defines the ability of an imaging system to distinguish between different wavelengths of light. The spectral resolution is defined by the number of spectral bands a sensor can capture and by how narrow or broad these bands are. For example, a greyscale image contains only one spectral band that often covers the entire visible part of the electromagnetic spectrum. Colour images typically divide the same part of the spectrum over three narrower bands, to obtain a separate measurement for red, green and blue reflectance. The spectral resolution can be increased further by measuring more and narrower spectral bands. Furthermore, bands outside the visible part of the spectrum can be added, for example in the infrared (IR), ultraviolet (UV), or X-Ray part of the spectrum.

The temporal resolution refers to the ability of a phenotyping system to capture data at frequent intervals over time. The temporal resolution of a phenotyping system depends on the acquisition time per plant and on the amount of plants a system needs to monitor. Temporal resolution is important to consider because plants are dynamic over time. By measuring at high temporal resolution, researchers can gain insight into plant development and the response of the plant to biotic or abiotic stress (Kjaer & Ottosen, 2015).

An important aspect for automated phenotyping systems is to bring the plant and the sensor together. This requires a system in which either the sensor is attached to a mobile platform that carries the sensors to the plant, or the plants are grown in a system that is able to transport the plants to the sensor. By bringing the sensor to the plant, measurements can be obtained in the growing environment of the plant. Sensor carriers for this purpose include robotic platforms that move through greenhouses, tractors or robotic field platforms, unmanned aerial vehicles (UAVs), and satellites (Yao et al., 2021).

1.2.3 Data analysis: from sensor data to plant measurements

Research to relieve the phenotyping bottleneck focused mostly on how sufficient and high-quality data could be acquired. With the increasing availability of systems that produce large volumes of sensor data, the phenotyping bottleneck was extended by the interpretation bottleneck (Smith et al., 2021). The interpretation bottleneck refers to the lack of computer algorithms to translate the raw sensor data into valuable information. It is beyond the scope of this introduction to provide a complete overview of computer-vision algorithms that are available for this purpose. Rather, the distinction is made between traditional computer-vision methods and methods based on deep learning. For a recent review on computer vision technologies for plant phenotyping we refer to the work of Li et al. (2020).

Traditional computer vision methods for plant phenotyping are based on a set of predefined rules to analyse images. Typically, hand-designed features are extracted based on colour, shape, and texture. These features are used to identify relevant parts of the image, from which measurements can be extracted, or for example to classify content of images as good or bad depending on a set of thresholds. Traditional computer vision methods can be effective when images are taken under controlled conditions, limiting the level of variation present in the dataset. However, generalisation of the algorithms to uncontrolled conditions or to other crops is often limited.

In contrast, supervised deep-learning based methods for plant phenotyping involve using deep neural networks that learn the features and patterns that are relevant for the task from a large training dataset. These methods can be trained to assess plants and their characteristics without the need for manual feature definition, allowing them to adapt to a wider range of plant species and conditions. However, the required datasets for training are labour-intensive to generate and the algorithms can be computationally intensive to train, requiring specialized hardware and expertise.

Overall, both traditional computer vision methods and deep learning methods have their strengths and limitations for plant phenotyping, and the choice of method depends on the specific needs and goals of the experiment or application.

1.3 Plant architecture of cucumber as a use case

In section 1.1, the need for large, high-quality phenotypic datasets was identified and the phenotyping bottleneck was introduced as one of the reasons that these datasets are often not available. Digital phenotyping was then proposed as a concept that could contribute to the solution of this problem. Section 1.2 introduced various approaches with respect to what part of the plant could be measured, what sensors are available, and what data analysis methods could be used for translating sensor data into plant measurements. To focus our research, in this section, we present the use case of plant architecture measurements in cucumber.

Cucumber is an important fruit-vegetable crop. Furthermore, the fast crop development and the relatively open structure of the crop, as compared to for example sweet pepper, were practical reasons to select cucumber as a model crop for our use case. Although measuring root systems is relevant, we focused on the above-ground parts of the plant. The architecture of the above-ground plant parts, or the 3D organisation of these plant parts, is defined by traits like plant height, leaf or branching angle, leaf size, and internode length. These traits are relevant for a number of reasons. First, the position and orientation of the leaves has an effect on the amount of intercepted light, which on its turn influences plant productivity. Second, developments in automated crop handling might lead to specific requirements with respect to the plant architecture and finally, changes in plant architecture over time can be an indicator of plant stress. (Najla et al., 2009; Paulus, 2019; Sibomana et al., 2013). Internode length is one of the traits that make up the plant architecture and was used as a model trait in this thesis. Nodes are points along the stem where leaves or branches are attached. Internode length is the distance along the stem between two consecutive nodes. By measuring internode length frequently, detailed information about the plant development over time can be obtained.

As plant architecture is about the 3D organisation of plant parts, it was expected that 3D data would be a more appropriate choice than 2D data. However, the state-of-the-art of 3D computer-vision methods, especially in the plant domain, is not as far developed as the state-of-the-art of 2D computer-vision methods. In this thesis, we aim to push forward the state-of-the-art in 3D computer-vision methods for plant phenotyping.

However, adding a third dimension does not only increase the potential level of information in the data, but also increases the complexity of the algorithms that have to be developed for analysing the data. Therefore, we started with the development of a method based on 2D data. The 2D and 3D methods were compared to each other, providing insight into the added value of 3D data for plant phenotyping.

In summary, this thesis explores digital phenotyping technology to develop methods for measuring plant-architectural traits. Cucumber was used as a model crop and internode length was used as a model trait. Furthermore, 3D sensor data and the corresponding deep-learning based methods to translate the 3D data into plant measurements were selected as promising technologies. The developed 3D-based methods were compared to a 2D-based approach, to gain insight into the added value of 3D data for digital plant phenotyping.

1.4 A phenotyping system to collect 2D and 3D data

To develop the methods identified in the previous section, a digital phenotyping system was required that could generate a dataset containing both 2D and 3D data of a population of cucumber plants. Therefore, in preparation for this PhD project, a custom phenotyping system was designed in collaboration with WIWAM phenotyping robots (WIWAM, 2022).

The phenotyping system was installed in a climate chamber and consisted of two platforms. The first platform drove in-between the plant rows and could be docked in the second platform. The second platform then moved the entire system from one plant row to the next. The platform that drove in-between the plant rows was equipped with a vertical scanning axis of 2.5m height, on which the sensors were mounted. An IMPERX B4820 16 MP CCD camera with an image resolution of 4904 x 3280 pixels (IMPERX, 2018) was used to collect 2D images. Furthermore, a Phenospex F500 dual scan system (Phenospex, 2017) was installed to collect 3D point clouds of the same plants. The Phenospex PlantEye F500 is a multispectral 3D scanner for plant phenotyping, which also provided spectral reflectance information (red, green, blue and NIR) for each point in the point clouds.

The data was collected in 2018. Twelve cucumber plants were grown in the climate chamber. The plants were placed on plant gutters with an in-row plant distance of 1 meter. This prevented that plants were occluding each other. The 2D images and the 3D point clouds were taken from the same plants, using multiple viewpoints per plant, over a period of 11 days. During this period, the plants grew from an average plant height of 76 cm to an average plant height of 195 cm. For specific details about the dataset and how it was collected we refer to the chapters in which the data was used. An example of the data is shown in Figure 1.2.

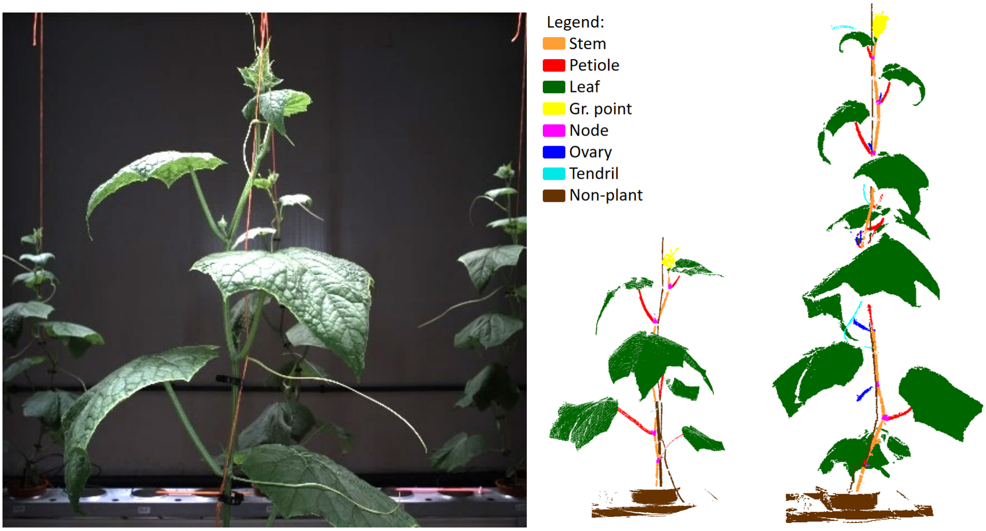


Figure 1.2 – An example of a 2D image (left) and an example of two 3D point clouds (right). The point cloud of the smaller plant was taken at the beginning of the data acquisition period, while the point cloud of the larger plant was taken at the end of the data acquisition period. The colours in the point cloud represent the different plant organs that were considered in chapters 3, 4, and 5, as specified in the legend.

1.5 Objective, main hypothesis, and thesis outline

The main objective of this thesis is to provide insight into the added value of 3D data and 3D-based methods for plant phenotyping of plant-architectural traits. To achieve this objective, methods based on 2D data and methods based on 3D data were developed and compared to each other. The main hypothesis that is tested in this thesis is:

> ***"phenotyping methods based on the acquisition and analysis of 3D data lead to improved phenotypic measurements of plant-architectural traits as compared to measurements obtained using 2D-based methods."***

This hypothesis was tested in the light of internode length, a model trait that is part of the plant architecture.

The first approach to measure internode length was based on the 2D dataset. In **chapter 2**, an automated digital phenotyping method to measure 2D internode length is presented. A deep convolutional neural network was trained to detect the nodes in the images. The nodes detected in images from multiple viewpoints around the plant were combined and the 2D internode length was then estimated as the Euclidean distance between two consecutive nodes. The estimated internode lengths were compared to two different reference measurements. The first reference measurement was based on a fast manual estimation of the internode length and the second reference measurement was obtained by measuring each individual internode with a measuring tape. Although our 2D digital method was more accurate than the fast manual method, the highest accuracy was obtained with the measuring tape.

One of the main limitations of the 2D method was that a fixed plant-camera distance had to be assumed for the conversion from pixels to mm. However, 25% of the plants showed a curved growing pattern, meaning that the estimated internode lengths for these plants were significantly less accurate. Therefore, a method to estimate internode length based on the 3D data was developed next.

Although the focus was on internode length, a fully segmented point cloud was considered valuable for future applications. Therefore, the focus was first on predicting for each point in the point cloud to which plant organ it belonged. **Chapter 3** introduces a method to segment the point clouds into plant-parts. In one of the experiments, the training data set was manually labelled twice and the agreement between the two training sets was used as a way to evaluate the quality of the training data. The design of the phenotyping experiment and the effect of the number of classes that had to be recognized was analysed in a second experiment. Overall, it was shown that our method provides a suitable base for point-cloud segmentation for plant phenotyping. Furthermore, the added value of spectral data for point-cloud segmentation was quantified and it was shown that the availability of spectral data significantly improved the segmentation quality.

One of the findings that is reported in chapter 3, is that the current segmentation task suffers from a class-imbalance problem. Therefore, in **chapter 4**, a method to improve the segmentation of the classes for which less data was available is presented. Inspired by methods available to handle class imbalance in 2D images, a method to focus the attention of the 3D neural network on the smaller classes was proposed and tested. This method allows to select classes of interest and focus the learning process of the neural network on those classes. By shifting the attention towards the class ‘node’, the segmentation quality was significantly improved.

The improved segmentation of the class ‘node’ from chapter 4, was then used in **chapter 5** to test if the improvement was sufficient to detect node-objects in the segmented point clouds. A clustering algorithm was used to group together points that belonged to the same node, moving from segmented points to plant parts (objects). Based on these detected nodes, the 3D internode length was estimated and compared to the 2D internode length estimates that were presented in chapter 2. Although the 2D method was able to detect more nodes, the estimated internode lengths were more accurate when using the 3D method.

Finally, in **chapter 6**, the conclusions of this thesis are summarized and a general discussion of the work presented is this thesis is provided. Following from the objective to provide insight into the added value of 3D data and methods for plant phenotyping, the thesis is concluded with a reflection on the question posed in the title of the thesis: Digital plant phenotyping in three dimensions – what’s the point?

Chapter 2

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

This chapter has been published as:

Boogaard, F. P., Rongen, K. S. A. H., & Kootstra, G. W. (2020). Robust node detection and tracking in fruit-vegetable crops using deep learning and multi-view imaging. *Biosystems Engineering*, *192*, 117–132. https://doi.org/10.1016/j.biosystemseng.2020.01.023

Nomenclature

A	Numbering accuracy [-]
B^{p}	Bounding box *p*
c	Completeness [-]
e_i	Error between estimated internode length and ground-truth internode length [mm]
$e_{ij,\ accurate}$	Error of the accurate manual method [mm]
$e_{ij,\ rough}$	Error of the rough manual method [mm]
h	Homogeneity [-]
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 [%]
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, FP, TN, FN	True positive, False positive, True negative, False negative
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]

Abstract

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 focuses 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.

2.1 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 et al., 2010). Modern breeding methods apply this knowledge to increase the genetic gain of the breeding process and accelerate the 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 et al., 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 (L. Li et al., 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 et al., 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 et al., 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 Midtiby (2017) proposed a method for weed detection in highly occluded cereal fields. For segmentation and counting of leaves, Giuffrida, Doerner, and Tsaftaris (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.

2.2 Materials & Methods

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

2.2.1 Experimental set-up

2.2.1.1 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 × 60) / 459 = 11 s. This is including time to move from plant to plant and to register the data on a laptop.

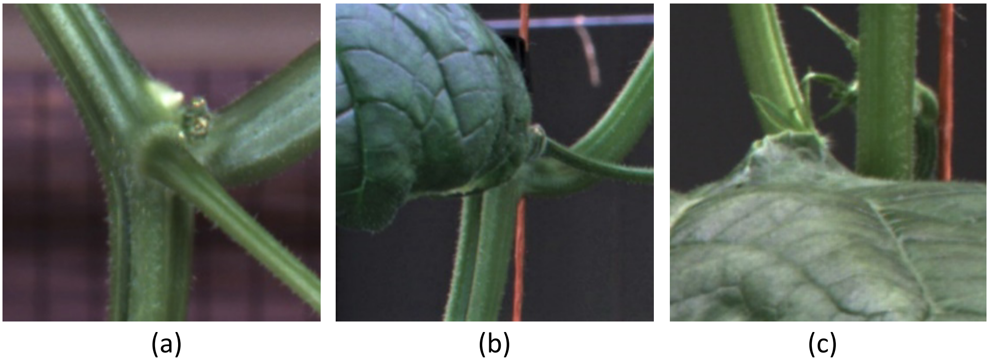
2.2.1.2 Camera set-up and image acquisition

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 x 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 Figure 2.1. Taking images from multiple viewpoints increases the chance of capturing the object of interest (in our case the node) of a plant (Hemming et al., 2014; Nguyen et al., 2016). Moreover, nodes can now be detected and combined in multiple images, improving the robustness of the system. Figure 2.2 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 to 6). This was done from both sides of the plant gutter (0° and 180°) resulting in 36 viewpoints per plant. In Figure 2.3, example images from one side of the plant gutter are shown. In section 2.2.2.2, the method to combine the different viewpoints is explained in more detail.

2.2.1.3 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 plants × 8 days × 4 runs × 36 images = 13,824 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 9,990 suitable images.



(a) (b) (c)

Figure 2.1 – (a) completely visible node, (b) partly occluded node and (c) fully occluded node.

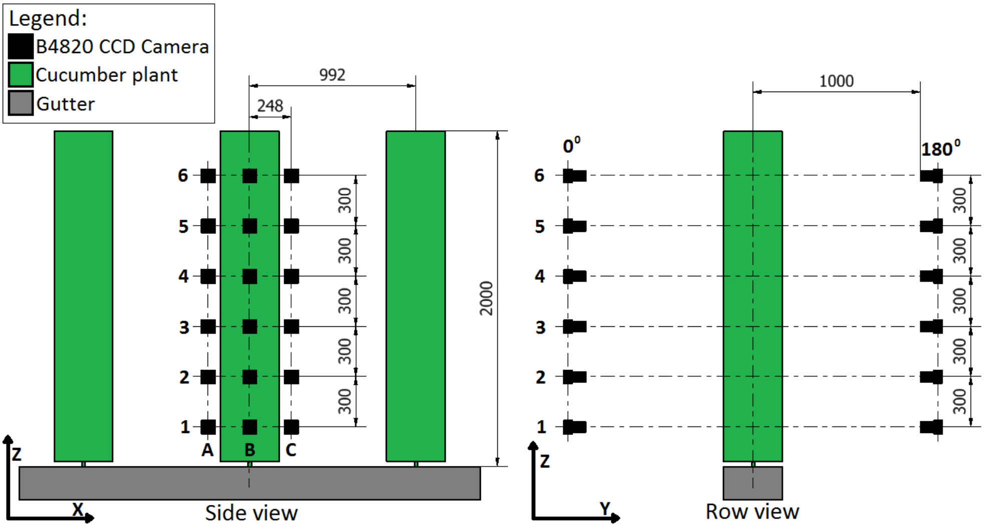


Figure 2.2 – Side view and row view of the camera positions used for image acquisition. The side view image shows the camera positions A, B, and C for the 6 height levels. In the row view image it is shown that images are taken from both sides of the plant gutter. The dimensions are in mm.

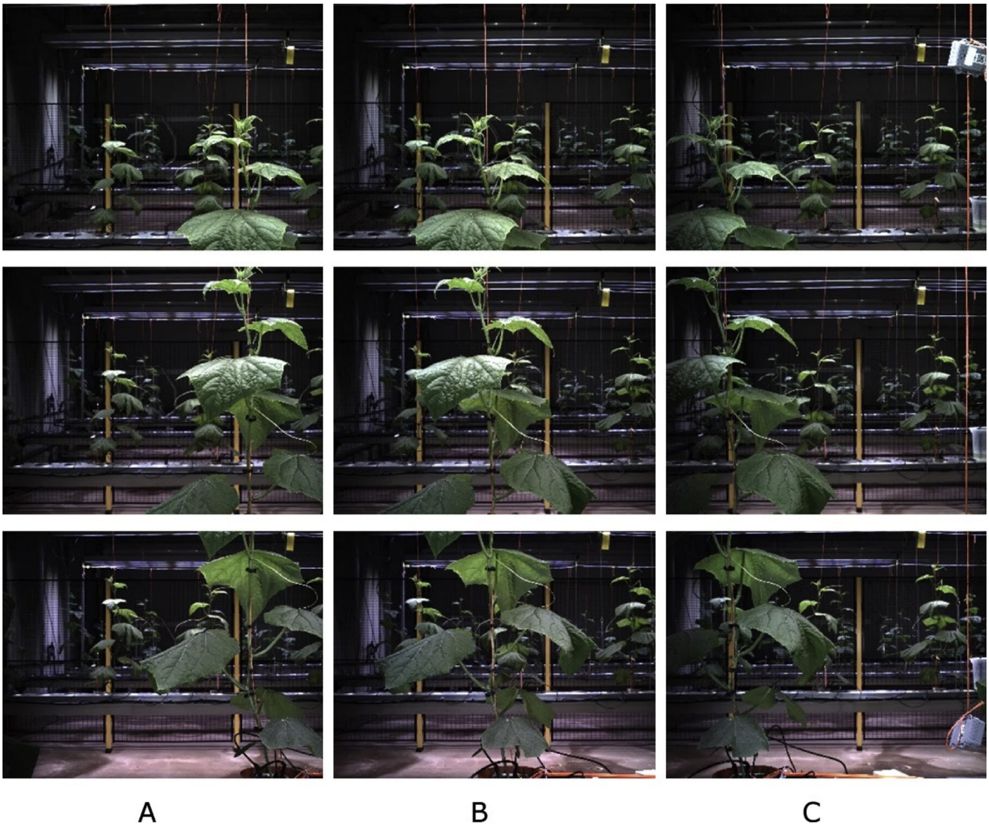


Figure 2.3 – Examples of captured images from one side of the plant gutter, three camera positions and three of the six height levels.

2.2.2 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.

2.2.2.1 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 et al., 2015).

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 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 et al., 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.

Training of the network

In order to train the network, a training set with manually annotated nodes was generated using the LabelImg 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 Figure 2.1. 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 8,877 fully visible nodes and 1,437 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 6,500 and 6,700 nodes and the validation dataset contained between 2,100 and 2,300 nodes.

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.

2.2.2.2 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 Figure 2.4a). 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 plane 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 Figure 2.4b.

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 2.2.1. 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.

2.2.2.3 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 internode 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.

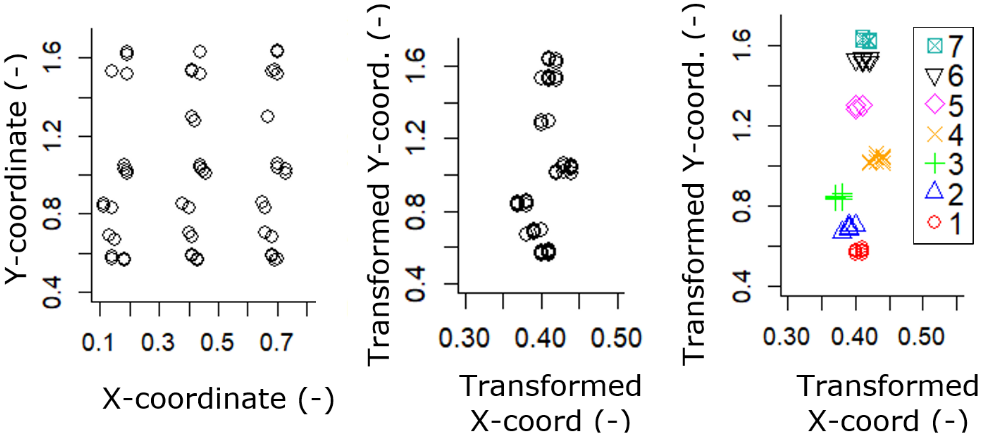
Commonly, AP is used for clustering data of which both the spatial distribution and the number of clusters are unknown. However, since in our case we assume (near) vertical plant growth, some information about the spatial distribution of the nodes is available and can be used to support the clustering 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 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 5-15 and q_y ranging from 19-29.

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 Figure 2.4.



(a) (b) (c)

Figure 2.4 – (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.

2.2.2.4 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 mm × 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 Equation 2.1.

$$\hat{s}_{ij}(t) = \sqrt{\left(x_j^{mm} - x_i^{mm}\right)^2 + \left(y_j^{mm} - y_i^{mm}\right)^2} \qquad \text{[mm]} \qquad \textit{Equation 2.1}$$

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.

2.2.3 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.

2.2.3.1 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 2.2.1.2) 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.

2.2.3.2 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 et al., 2010). The IOU value is calculated by dividing the area of the intersection of the two bounding boxes by the area of their union:

$$IOU(B^p, B^a) = \frac{|B^p \cap B^a|}{|B^p \cup B^a|} \qquad [-] \qquad \textit{Equation 2.2}$$

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 $\mathrm{IOU}(B^{\mathrm{p}}, B^{\mathrm{a}}) > 0.5$, 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 Equation 2.3. The recall describes how many of the real nodes are detected by the network, as specified in Equation 2.4. The F1-score defined in Equation 2.5 calculates the harmonic mean of the precision and recall, which allows to compare the performance based on a single indicator (Sasaki, 2007).

$$\text{Precision} = \frac{\#\ \text{TP}}{\#\ \text{TP} + \#\ \text{FP}} \qquad [\text{-}] \qquad \textit{Equation 2.3}$$

$$\text{Recall} = \frac{\#\ \text{TP}}{\#\ \text{TP} + \#\ \text{FN}} \qquad [\text{-}] \qquad \textit{Equation 2.4}$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \qquad [\text{-}] \qquad \textit{Equation 2.5}$$

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 Figure 2.1b) 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.

2.2.3.3 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 Equation 2.6.

$$V = 2 \cdot \frac{h * c}{h + c} \qquad [-] \qquad \textit{Equation 2.6}$$

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 Equation 2.7. The value NP^{c} was determined by combining the predicted node order resulting from Step 3 of our method (section 2.2.2.3) with the annotated node order.

$$A = \frac{\mathrm{NP}^{\mathrm{c}}}{\mathrm{NP}} \qquad [-] \qquad \textit{Equation 2.7}$$

2.2.3.4 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 $i+1$ (Equation 2.8). 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 Equation 2.9 to enable comparison to the results of Yamamoto et al. (2016) and Nguyen et al. (2015).

$$e_i = \hat{s}'_i - s_i \qquad [\mathrm{mm}] \qquad \textit{Equation 2.8}$$

$$r_i = \frac{|e_i|}{s_i} \qquad [\%] \qquad \textit{Equation 2.9}$$

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 Equation 2.10. 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 Equation 2.11. In both calculations, only the first five internodes were taken into account, to prevent internodes still elongating from influencing the results.

$$e_{ij,\ rough} = s_{ij} - \frac{\sum_{i=1}^{n} \sum_{j=1}^{d} s_{ij}}{d * n} \qquad \text{[mm]} \qquad \textit{Equation 2.10}$$

$$e_{ij,\ accurate} = s_{ij} - \frac{\sum_{j=1}^{d} s_{ij}}{d} \qquad \text{[mm]} \qquad \textit{Equation 2.11}$$

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).

2.3 Results

Section 2.3.1 presents the analysis of the number of viewpoints required for successful node detection. The results of the node-detection algorithm are presented in section 2.3.2. In section 2.3.3, 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 2.3.4.

2.3.1 Multiple viewpoints

The average number of observations of a node as a function of the number of viewpoints is plotted in Figure 2.5. 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.

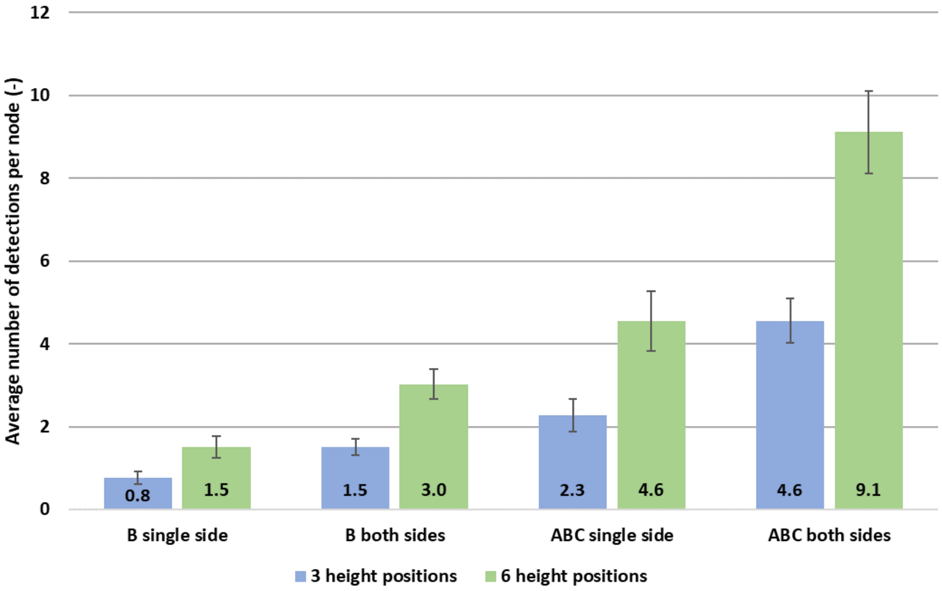


Figure 2.5 – Average number of times a particular node was detected by the algorithm for different viewpoint configurations. The error bars show the standard deviation.

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 2.6 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.

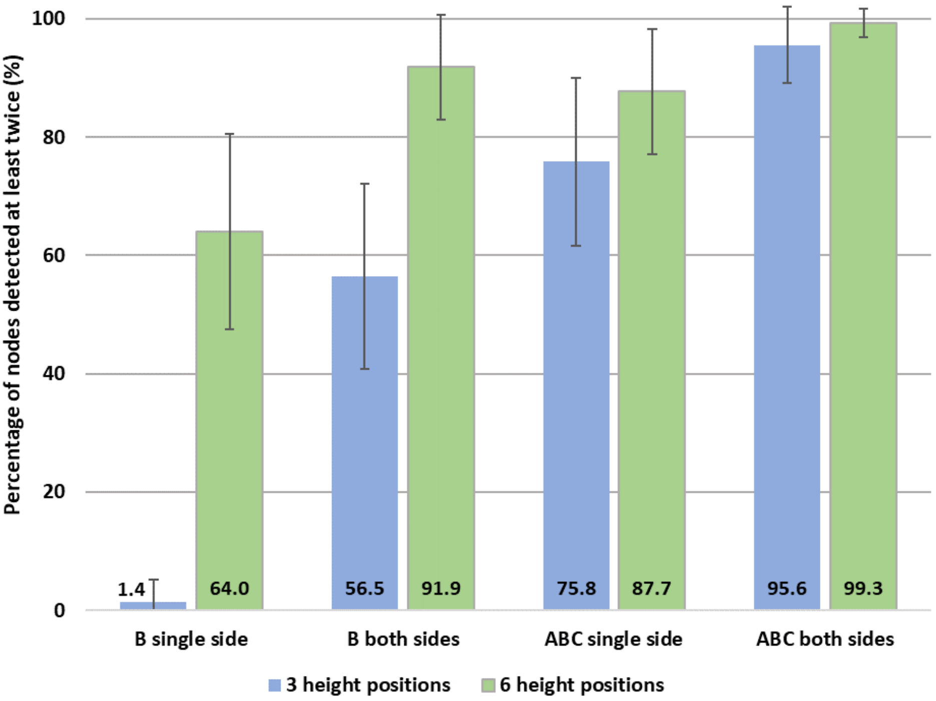


Figure 2.6 – Percentage of nodes detected at least twice for different viewpoint configurations. The error bars show the standard deviation.

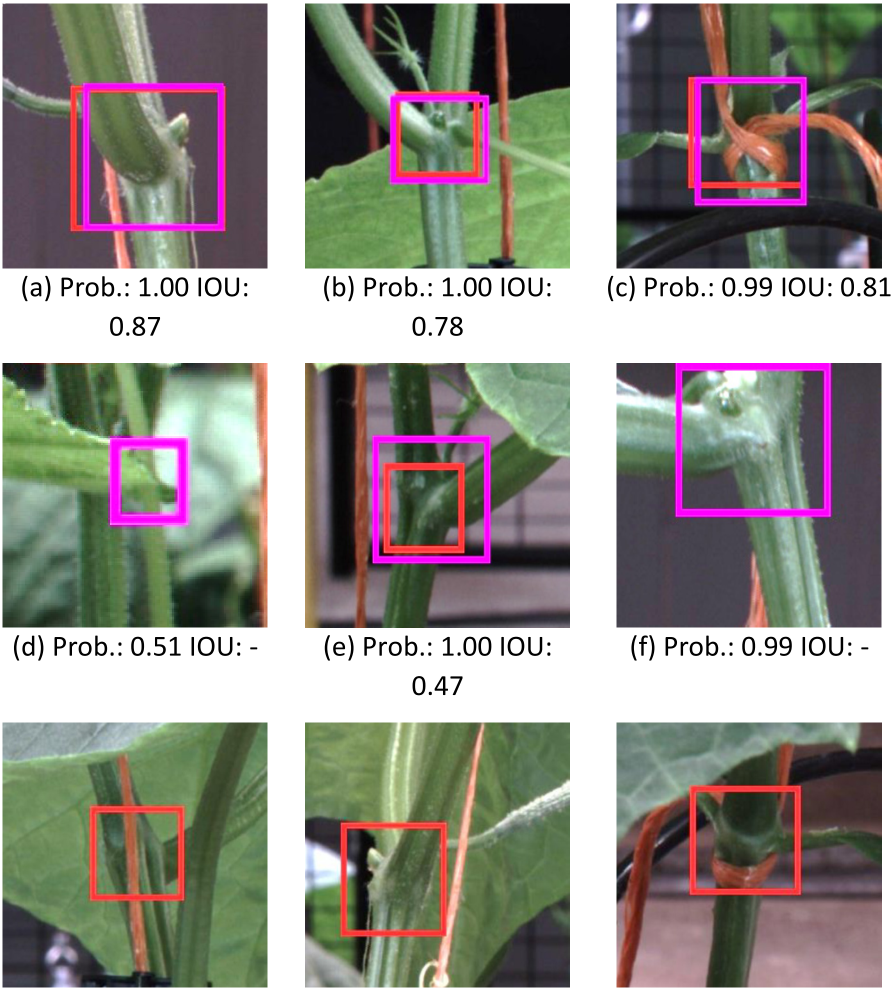
2.3.2 Node detection (Step 1)

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 (σ=0.005), 0.99 (σ=0.0) and 0.98 (σ=0.004) respectively. For the validation set, the precision, recall and F1-score were 0.91 (σ=0.01), 0.92 (σ=0.01) and 0.92 (σ=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 (σ=0.01), 0.92 (σ=0.01) and 0.94 (σ=0.01), as presented in Table 2.1.

Table 2.1 – 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**

To obtain more insight into the detections of the algorithm, Figure 2.7 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 Figure 2.7 shows three typical false positive failures of the node-detection algorithm. Figure 2.7d 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 Figure 2.7e 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 Figure 2.7f, 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 Figure 2.7 shows three false negative examples. These nodes were annotated in the ground truth data, but not detected by the algorithm.



(a) Prob.: 1.00 IOU: 0.87

(b) Prob.: 1.00 IOU: 0.78

(c) Prob.: 0.99 IOU: 0.81

(d) Prob.: 0.51 IOU: -

(e) Prob.: 1.00 IOU: 0.47

(f) Prob.: 0.99 IOU: -

(g) Prob.: - IOU: -

(h) Prob.: - IOU: -

(i) Prob.: - IOU: -

Figure 2.7 – 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).

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 (σ=0.06). This is considerably lower than the average recall obtained on all nodes of 0.92 (σ=0.01), indicating that it was more difficult to find the youngest node of a plant.

2.3.3 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 2.2. 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 2.2 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 2.2.1.1, 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 2.2 show that the clustering and numbering algorithm were able to cluster the annotated bounding boxes almost identical to the annotated node numbering when excluding the outlier plants from the analysis.

Table 2.2 – 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
Excl. outliers	0.98	0.03	0.99	0.02	0.98	0.02	0.97	0.03

In Table 2.3, 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.

Table 2.3 – 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
Excl. outliers	33.3	65.3	1.4	0.0

2.3.3.1 Parameter check node clustering and cluster numbering

The distribution of the performance measurements obtained from the grid search, introduced in section 2.2.2.3, is presented in Table 2.4. 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 5-15 and q_y ranging from 19-29 in the second clustering step. The results are again presented in Table 2.4. 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.

Table 2.4 – 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.

Grid		**Node clustering optimisation**			**Numbering**
search:		**Homogeneity**	**Completeness**	**V-measure**	**accuracy**
step 1	Min.	0.62	0.74	0.75	0.11
	Max.	0.99	1.00	0.99	0.98
	Mean	0.87	0.89	0.87	0.44
	St. dev.	0.02	0.02	0.02	0.10
step 2	Mean	0.98	0.99	0.98	0.97
	St. dev.	<0.0001	<0.0001	<0.0001	<0.0001

2.3.4 Complete pipeline (Step 1-4)

2.3.4.1 Estimation of internode length

In Figure 2.8, 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 Figure 2.8. 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 (Equation 2.9) 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.

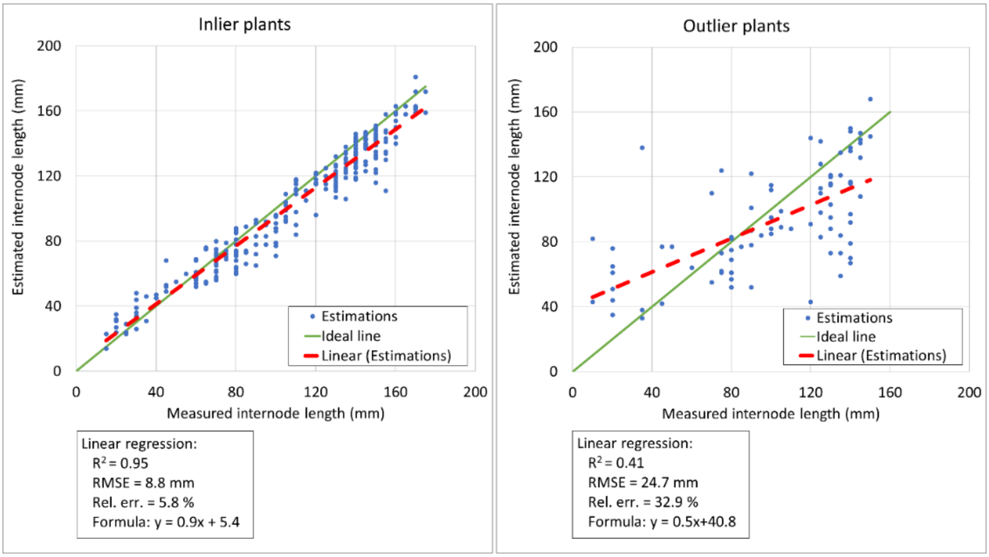


Figure 2.8 – The relation between the estimated internode length and the manually measured internode length specified for inliers (plants having a (near) vertical growth pattern) and outliers.

The error in the calibrated estimation of the internode length for all plants (Equation 2.8) is shown in Figure 2.9. 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 to 5.1 mm and the lower and upper quartiles range from -10.0 mm to -2.1 mm and 2.2 mm to 10.5 mm respectively. For the outlier plants, the medians are in the range of -3.7 mm to 12.9 mm and the lower and upper quartiles range from -36.4 mm to -8.4 mm and 8.0 mm to 20.1 mm respectively.

2.3.4.2 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 2.2.3.4. Figure 2.10a 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 Equation 2.11. The rough manual measurement had a 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.

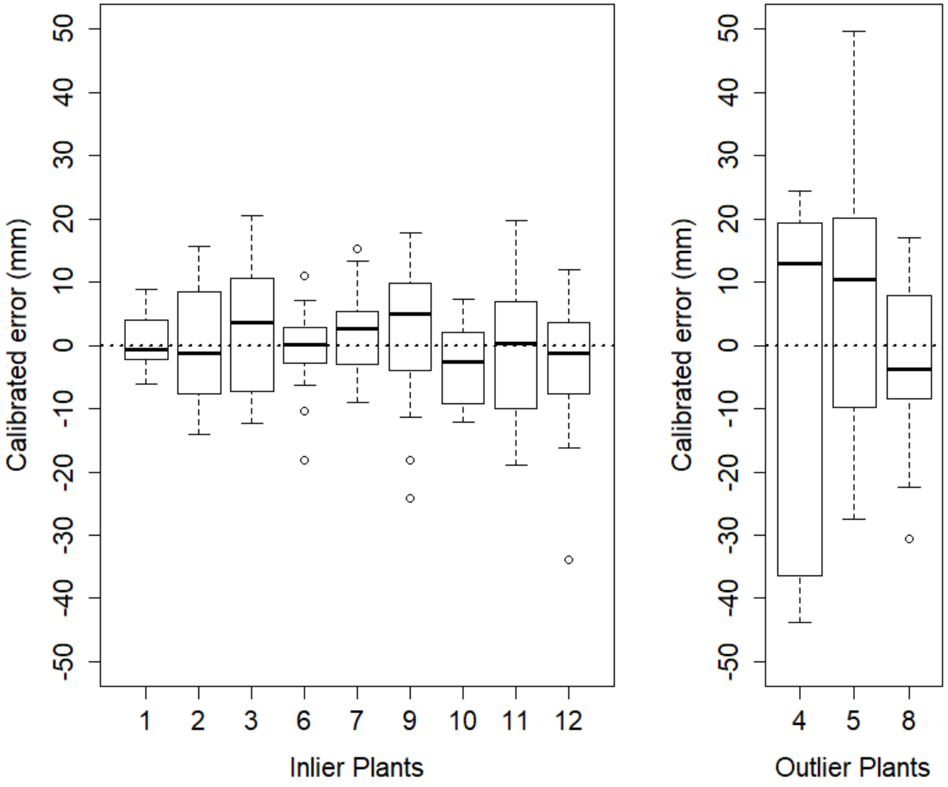
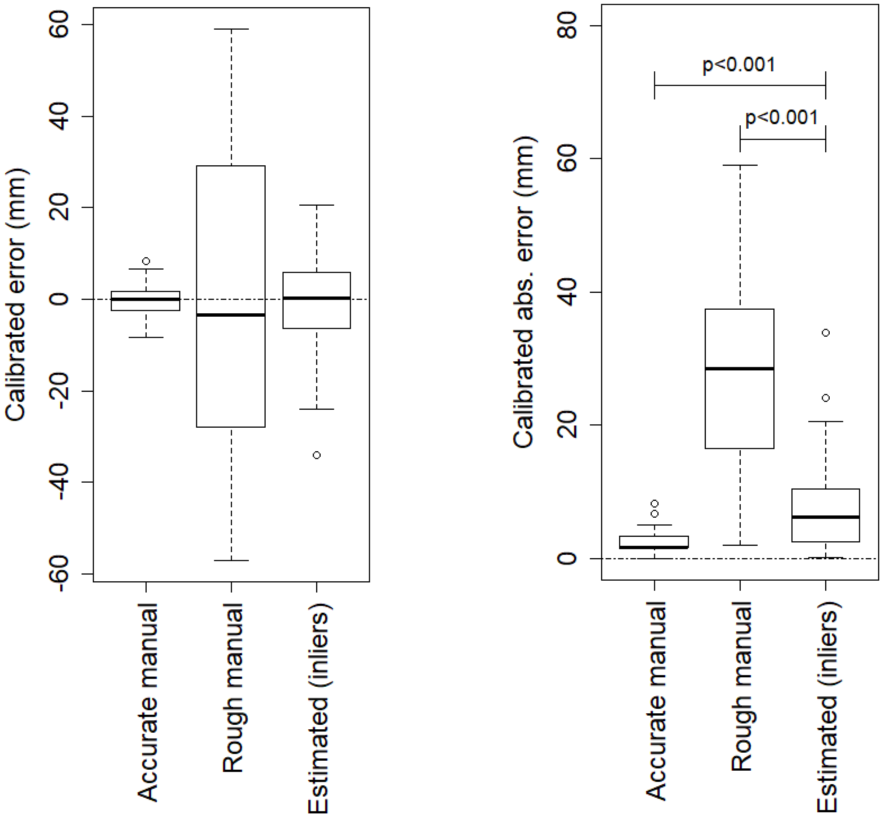


Figure 2.9 – Box-and-whisker plot of the error (Equation 2.8) of the inlier (left) and the outlier (right) plants. The thick black line indicates the median, the box shows the lower and upper quartile, and the whiskers indicate the highest and lowest error, where points outside the interquartile range are considered outlier values (noted as a circle).

The absolute errors of the different methods are shown in Figure 2.10b. 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.



(a) Relative calibrated errors

(b) Absolute calibrated errors

Figure 2.10 – Box-and-whisker plots comparing the two manual measurement methods, accurate and rough, with the estimations from our automatic method.

2.4 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.

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 2.7e). 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 et al. (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, 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.

2.5 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.

Chapter 3

Boosting plant-part segmentation of cucumber plants by enriching incomplete 3D point clouds with spectral data

This chapter has been published as:

Boogaard, F. P., van Henten, E. J., & Kootstra, G. (2021). Boosting plant-part segmentation of cucumber plants by enriching incomplete 3D point clouds with spectral data. *Biosystems Engineering*, *211*, 167–182. https://doi.org/10.1016/j.biosystemseng.2021.09.004

Nomenclature

XYZ	Geometric features (x, y, z location of points)
RGB	Spectral features (red, green, and blue reflectance)
XYZRGB	The combination of geometric and spectral features
$\mathrm{TP}, \mathrm{FP}, \mathrm{TN}, \mathrm{FN}$	True positive, False positive, True negative, False negative
IoU_c, $\mathrm{IoU}_{\mathrm{micro}}$, $\mathrm{IoU}_{\mathrm{macro}}$	Intersection-over-Union for class c, or as micro or macro average

Abstract

Plant scientists require high-quality phenotypic datasets. Computer-vision based methods can improve the objectiveness and the accuracy of phenotypic measurements. In this paper, we focus on 3D point clouds for measuring plant architecture of cucumber plants, using spectral data and deep learning. More specifically, the focus of this paper is on the segmentation of the point clouds, such that for each point it is known to which plant part (e.g. leaf or stem) it belongs. It was shown that the availability of spectral data can improve the segmentation, with the mean intersection-over-union rising from 0.90 to 0.95. Furthermore, we analysed the effect of uncertainty in the collection of ground truth data. For this purpose, we hand-labelled 264 point clouds of cucumber plants twice and show that the intra-observer variability between those two annotation sets can be as low as 0.49 for difficult classes, while it was 0.99 for the class with the least uncertainty. Adding the second set of hand-labelled data to the training of the network improved the segmentation performance slightly. Finally, we show the improved performance of a 4-class segmentation over an 8-class segmentation, emphasizing the need for a careful design of plant phenotyping experiments. The results presented in this paper contribute to further development of automated phenotyping methods for complex plant traits.

3.1 Introduction

Plant breeding has improved crop varieties with respect to for example yield, stress resistance and plant and fruit morphology for a long time. The effectiveness of the breeding process can be further improved by studying the relationship between genotype, phenotype and environment (Houle et al., 2010) for these aspects. These studies require high quality genotypic, phenotypic and environmental datasets (Yol et al., 2015). Current phenotypic measurements, however, are mainly based on human observation and therefore tend to be subjective and descriptive, limiting the quantity and the quality of the observations (Gehan & Kellogg, 2017). In contrast, computer-vision techniques can be used to obtain phenotypic measurements in an automated and more objective way.

Traditionally, most computer-vision based phenotyping research applies feature-based machine learning algorithms, where the algorithms to extract image features are designed by hand. Especially in a complex and cluttered greenhouse environment, this is a challenging task, because of the high level of variation present in the data (Minervini et al., 2015). First, there is the intrinsic variation in shape and appearance between plants and plant parts, within a cultivar and between cultivars. Additionally, there is an extrinsic source of variation caused by different growing environments (e.g. lighting conditions or planting density). The environmental variation also includes variation due to differences between growing systems and crop maintenance (Lobos et al., 2017). In contrast to using hand-designed features, in deep-learning-based methods, the features are learned together with the decision making from labelled data in one integrated deep neural network. This allows joint optimization of feature extraction as well as decision making. In current state-of-the-art, deep-learning-based methods outperform feature-based methods. Computer-vision techniques based on deep learning seem to be a promising tool to cope with the challenges present in the plant domain. (Pound et al., 2017; Ubbens & Stavness, 2017)

In this paper, we focus on computer-vision based methods to collect phenotypic data about the plant architecture of cucumber plants. Plant architecture is the set of phenotypic traits defining the three-dimensional (3D) organisation of the plant parts (Reinhardt & Kuhlemeier, 2002). High quality phenotypic data about the plant architecture supports plant breeders in their efforts to optimise plant architecture and it supports and informs growers on the evaluation of their crop balance and decision making with respect to plannable crop activities. Being able to adapt the plant architecture also allows to work towards crops that are optimised for automated harvesting or other crop maintenance tasks. Because of the complex structure of cucumber plants, we focused on 3D computer-vision methods. This follows the recommendation of Boogaard, Rongen and Kootstra (2020) based on the limitations of two-dimensional (2D) computer-vision for internode length measurements.

An important first step in developing 3D computer-vision methods, is to segment the input data into individual plant organs (Shi et al., 2019; Vázquez-Arellano et al., 2016). The segmented data in which all plant parts are known can then be used to develop methods to obtain phenotypic measurements, as was for example shown by Golbach et al. (2016). A recent overview of methods for segmenting 3D data has been presented by Guo et al. (2020). PointNet++ (Qi et al., 2017) was one of the top performing point-based methods identified in this work, although not specifically focusing on plant materials. In another recent comparison of point-based segmentation using deep learning (Turgut et al., 2020), the focus was on segmentation of rosebush plants. In their work, six recent 3D point-based segmentation methods were compared. The best segmentation results were obtained by PointNet++. Since PointNet++ was found to be one of the best performing methods in both reviews, we build on this method in our research.

Contributions of the paper

In the work of Shi et al. (2019), a method is presented to segment 2D plant images and project the segmentation into 3D space, to obtain a segmented 3D plant model. The method works well, but was only tested on small plants due to limitations in PointNet++. In this paper, we study the performance of a PointNet++-based segmentation method for large cucumber plants. The plants were grown in rows and scanned by two laser-triangulation sensors, resulting in a coloured 3D point cloud of the plant, partially incomplete due to a limited number of viewpoints. We split the point clouds into blocks that could be processed by PointNet++ without reducing spatial resolution. After testing the baseline performance of the segmentation, the segmentation method was further investigated in three experiments.

In the first experiment, we investigated the benefit of using spectral data in addition to the geometric data. In 2D computer-vision for plant phenotyping, it is common to use colour images (Dutagaci et al., 2020). These images provide information on the structure of objects in 2D, as well as on their spectral properties. Although in the review of Guo et al. (2020) examples are identified of deep-learning-based 3D segmentation methods using spectral data, the added value of spectral data for plant phenotyping has not been studied. Especially when working directly on point clouds, often only geometric data is available, as for example in the work of Turgut et al. (2020). We quantified the added value of spectral data for the purpose of plant-part segmentation by enriching the 3D point clouds with spectral data and comparing the segmentation results of the enriched data to the original performance.

In the second experiment, two manual annotations of the data were used to get more insight in the effect of the quality of the training data by studying the intra-observer variability and the benefit of adding multiple annotations in the training process. Generating this training data is challenging and partly an ambiguous process (Griffiths & Boehm, 2019a).

In the final experiment, it was considered that plant parts that are not relevant for quantifying the plant architecture could have been removed during crop maintenance. Furthermore, non-plant elements like the plant gutter could be given a distinct colour. These two changes in the experimental set-up could simplify the segmentation task. We simulated these effects in our data and tested the effect on the segmentation performance.

3.2 Materials & Methods

This section presents the data used for this research, including the plant materials, the data acquisition and the annotation of the data in section 3.2.1. The segmentation method is introduced in section 3.2.2, including a description of how we processed our data. In section 3.2.3, the evaluation criteria used to measure the performance of our method are introduced. Finally, in section 3.2.4, the experiments performed to answer the research questions are explained in more detail.

3.2.1 Data

3.2.1.1 Plant materials

Twelve plants of the cucumber variety Proloog RZ F1 (Rijk Zwaan, De Lier, The Netherlands) were grown in a climate chamber. The plants were attached to a supporting wire in order to grow vertically. To prevent occlusion between plants, the distance between each plant was 1 meter. The plants were monitored during 11 consecutive days starting at June 25th, 2018 and ending on July 5th, 2018. At the start of the experiment, there were 7 to 9 leaves per plant, which increased to 11 to 13 leaves at the end of the experiment. A few images of different plant parts of the cucumber plants are shown in Figure 3.1.

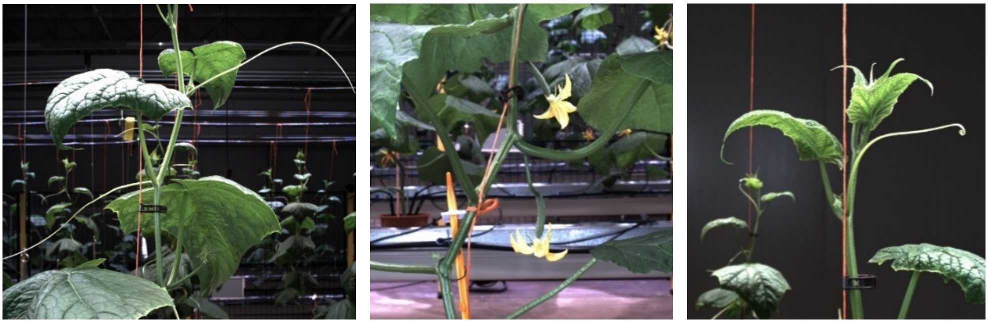


Figure 3.1 – Examples of different parts of the cucumber plants.

3.2.1.2 Data acquisition

The point cloud data for this research was generated using two Phenospex PlantEye F500 multispectral laser scanners, capturing 3D data based on laser-line triangulation. Besides the geometric data in the 3D point clouds, the scanners also provided the reflectance in red, green, and blue. The spectral measurements of the PlantEye F500 are independent of lighting conditions in the climate chamber. For this research, we refer to the geometric data as XYZ and to the spectral data as RGB.

The two scanners were mounted on an automated data acquisition system that can move through a climate chamber (see Figure 3.2). Since the scanners were mounted in a fixed frame with respect to each other, the point clouds from the individual scanners could be combined into one point cloud. To do this, the two point clouds of one of the scans were manually aligned. The resulting transformation matrix was then used to align all other scans of the cucumber plants during the experiments.

The data acquisition system was positioned to scan the plants from a side-view perspective from both sides of the plant gutter. At each position, the plant was scanned in a vertical direction, from bottom to top. The time between the scans from both sides of the plant was often more than 30 minutes. Due to (small) movements of the plant in this time period, we did not merge the scans from both sides of the plant gutter. This resulted in a total of $12\ \text{plants} * 2\ \text{sides} * 11\ \text{days} = 264\ \text{point clouds}$. A schematic overview of the layout of the climate chamber, including the scan positions and plant IDs, is shown in Figure 3.2.

The point clouds obtained by the data acquisition system were incomplete. This is due to occlusion by other plant parts and due to the horizontal position of the scanners, which resulted in the inability to measure horizontal surfaces (see Figure 3.3).

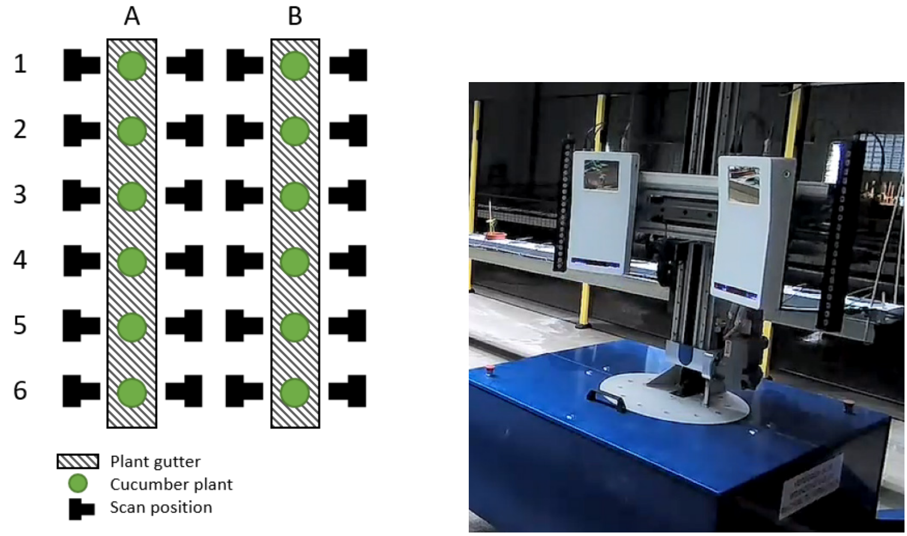


Figure 3.2 – Left: Top-view scheme of the 12 plants and the 24 scan positions. The rectangles show plant gutter A and plant gutter B, each containing six plants. Plants are identified as A1 to B6. Right: The point cloud acquisition system, showing the two PlantEye F500 scanners. The mobile blue platform was moved to the 24 scan positions indicated in the schematic overview. At each of these positions, the scanners were moved along the vertical axis to scan the plant from the pot to the growing point.

3.2.1.3 Data annotation

The training data for the network was obtained by manually labelling all of the 264 point clouds. This was done using the segment module of CloudCompare (CloudCompare, 2019). One class at a time, all points belonging to that class were selected and stored as a separate point cloud. For 2.3% of the points it was not clear to which class they belonged. These points were removed from the point cloud. The remaining points were divided in the classes stem, petiole, leaf, growing point (gr. point), node, ovary (flower and emerging fruit), tendril and non-plant. The class non-plant contained the plant gutter and construction, the pot and the supporting wire to which the plants were attached. An example of a manually segmented point cloud is shown in Figure 3.3.

At the transition from one class to another, it was sometimes difficult to define the exact boundary between different plant parts. For example, the boundary between node and stem and between node and petiole, or the attachment of the leaf to the petiole. Having incomplete data further increased the complexity of labelling the data by hand. To quantify the uncertainty of the annotations, all data was annotated twice. This was done by the same annotator. The second annotation was obtained approximately two weeks after the first annotation. The two annotated datasets are referred to as annotated dataset A (first annotation round) and annotated dataset B (second annotation round). The experiment in which the intra-observer variability between these two annotation sets is determined is explained further in section 3.2.4.2.

An overview of the distribution of the different plant parts in the two annotated datasets is shown in Figure 3.4. It is clear that the leaf (83.70%) and the non-plant material (9.49%) are overrepresented in the data, leading to an imbalanced dataset. The stem and petiole take up 2.80% and 1.88% of the points respectively. The number of points for each of the remaining classes (growing point, node, ovary and tendril) is less than 1% of the total number of points in the dataset.

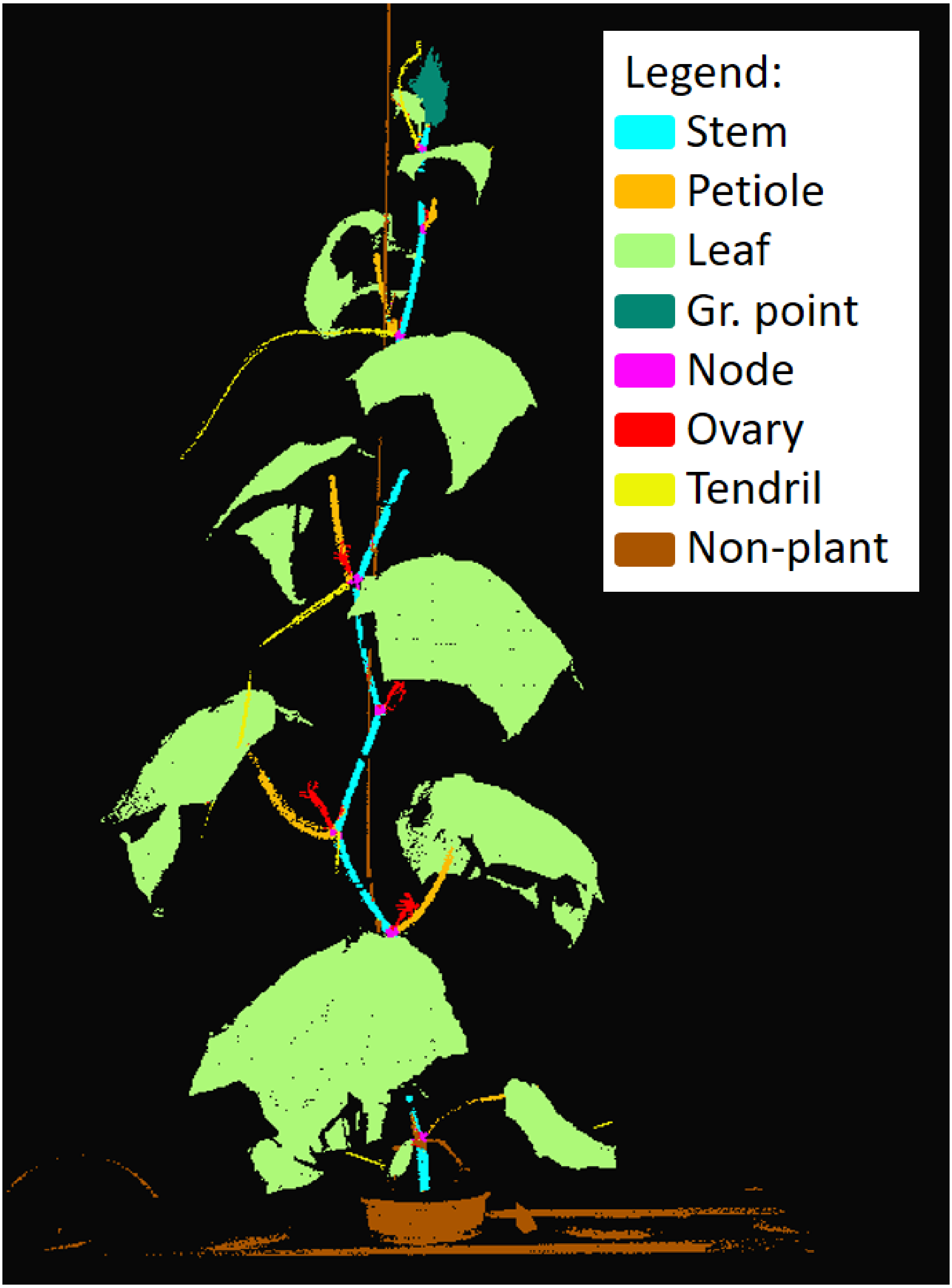


Figure 3.3 – Example of a manually segmented point cloud. Note the missing parts in the point cloud.

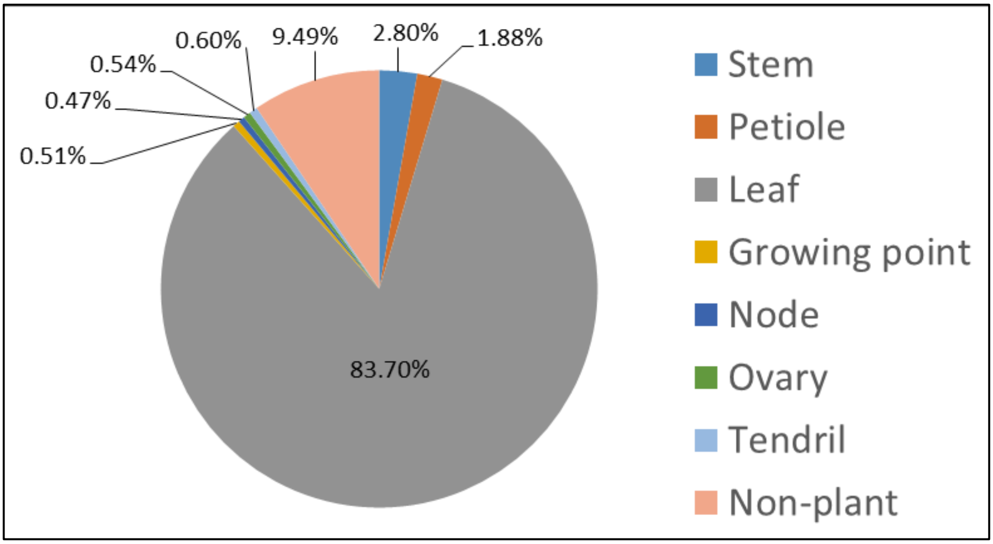


Figure 3.4 – Distribution of points over the different plant parts.

3.2.1.4 Data split

The annotated data was split in a training, validation and test set. This division was made on a plant level, to keep the data sets independent, by preventing that data from one plant was in more than one of the subsets. We randomly selected plant B6 for the validation set and plant A1 for the test set. All other plants were part of the training set. Although only one plant was used for the validation and test set, the validation set still contained 218 point clouds and the test set contained 258 point clouds. These numbers were obtained because of the division of the point clouds into separate blocks (see section 3.2.2.1) and because each plant was scanned from multiple sides on multiple days. The training set contained 2626 point clouds.

To quantify the variation in performance for different splits of the dataset, a cross-validation was performed. In this experiment, the training of the neural network was repeated for three other combinations of training, validation and test set. The used plant IDs (as defined in Figure 3.2) for these data splits are given in Table 3.1. The cross-validation was only done for experiment 1, in which the effect of adding spectral data was evaluated. The results are shown together with the other results for this experiment in section 3.3.1. For the other experiments, split 1 was used.

Table 3.1 – Division of plants over the training, validation and test set for the four cross validation runs.

	Train		Validation		Test	
Split	**Plant IDs**	**# Blocks**	**Plant ID**	**# Blocks**	**Plant ID**	**# Blocks**
1	A2-A6, B1-B5	2626	B6	218	A1	258
2	A1-A4, A6, B1-B2, B4-B6	2558	A5	252	B3	292
3	A1-A2, A4-A6, B1-B3, B5-B6	2551	B4	280	A3	271
4	A1-A6, B3-B6	2563	B2	265	B1	274

3.2.2 Point cloud segmentation

3.2.2.1 Data pre-processing

The number of points in a point cloud obtained using the presented data acquisition system was not constant and ranged from about 200,000 points up to 700,000 points, depending on plant size. However, PointNet++ requires a fixed number of points as input. Besides the varying number of points, it was not feasible to process an entire point cloud at once due to memory limitations. Therefore, the point clouds were divided into overlapping blocks of 40,000 points, similar to how entire scenes were processed by Qi et al. (2016) and how rosebush plants were processed by Turgut et al. (2020). To obtain a lower and more uniform point density resulting in bigger parts of the plant in each block, first, a voxel grid filter was applied. This filter was implemented in Point Cloud Library (Rusu & Cousins, 2011) and a voxel size of 2 * 2 * 2 mm was used, as a balance between resolution and memory usage.

The procedure to divide point clouds into blocks is described in pseudo code in algorithm 3.1. The filtered point cloud and the number of points per block are input arguments to the function. After initialization, the points are first sorted (line 9) along the z-axis (the vertical axis), such that the first block, containing the first *n* points, starts at the plant gutter. The blocks are generated in line 11-15 of the algorithm. The second block has an overlap of 50% with the first block and therefore contains points *n/2* to *(n/2) + n* (in our case 20,000 to 60,000). Blocks are formed until less than *n* points remain. For the last block (line 17-19), the last *n* points are taken from the point cloud, meaning that the overlap can be more than 50% with the second to last block. For small plants, it could happen that the entire point cloud contained less than *n* points. In that case, the while-loop (line 11-15) is skipped. After applying algorithm 3.1 to the original point clouds, the number of blocks per point cloud varied between 2 and 10, depending on plant size. The number of blocks for each of the point clouds is given in Table 3.A1 in the Appendix. An example of a small and a large point cloud and the division in blocks is shown in Figure 3.5.

Algorithm 3.1 – pseudo code to divide point cloud into blocks

```
split_pc_in_blocks(pc, n):
    # pc: original point cloud
    # n: number of points per block

    N = nr_points(pc)
    i0 = 0
    i1 = n
    block_list = []
    pc_sort = sort_on_z_value(pc)

    while i1 < N:
        new_block = pc_sort[i0:i1]
        block_list.append(new_block)
        i0 += n/2
        i1 += n/2

    # Add the last points
    last_block = pc_sort[N-n:N]
    block_list.append(last_block)
    return block_list
```

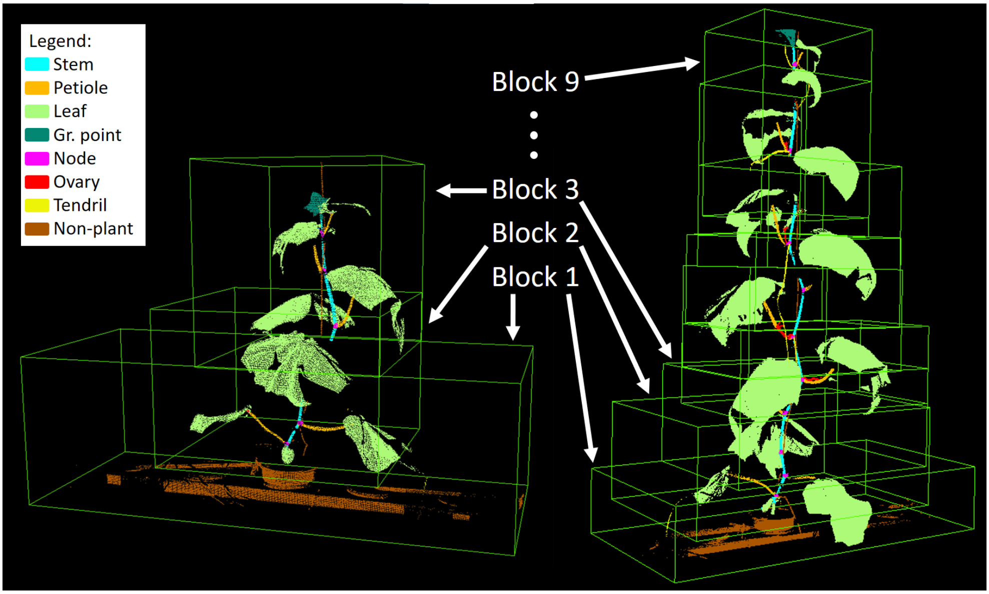


Figure 3.5 – Example of two point clouds divided into blocks. The point cloud on the left was obtained on the second measurement day and is divided into three blocks, the point cloud on the right was obtained on the tenth measurement day and is divided into nine blocks. The colours of the points represent the labels obtained by manual annotation. Block 1 contains the lowest part of the point cloud, including the plant gutter. The blocks have 50% overlapping points. All blocks contain n = 40,000 points. Note that because of the variation in width of the point cloud at different heights and the variation in point density, each block has its own dimensions.

3.2.2.2 Point cloud segmentation network

The neural network architecture PointNet++ as proposed by Qi et al. (2017) was trained to segment the blocks of the plants into plant parts. This network is an extension of the original PointNet (Qi et al., 2016), which was one of the pioneering deep learning methods to work directly on point clouds. In this work, we used the implementation published online by Qi et al. (2018). PointNet++ was designed as a hierarchical network to learn features on different scales in so-called set abstraction layers. Each of these layers consists of three steps. First, in the sampling layer, a set of points is selected from the input cloud to serve as centre points of the local regions, based on the XYZ coordinates. The grouping layer then finds groups of points in the neighbourhood of these centre points. Finally, a PointNet layer, based on the original PointNet architecture, is applied to learn features per group of points. The learned features are propagated to the points that were not sampled through distance-based interpolation. This is repeated in the other set abstraction layers to obtain features having larger receptive fields. Since the architecture of the network was not changed for this research, we refer to (Qi et al., 2018) for more details.

The input dimension of PointNet++ is *n * (d + c),* where *n* represents the number of points. As mentioned in the previous section, we have set the value for *n* at 40,000 points. For each point, there are *d + c* features, where *d* are the three coordinates of the point (*x, y* and *z*) and *c* are all the other features. In this research, $c = 0$ for the case where only geometric data was used and $c = 3$ for the case where RGB data was added. The output of the network is a probability matrix of *n* times the number of classes, showing for each point the probability that it belongs to these classes. Per point, the maximum class probability is selected as the predicted class.

A sparse softmax cross entropy loss is used to train the network. The learning rate of the network was set to 0.001. The network was trained until the loss on the validation set no longer decreased. The weights obtained at this point were used to evaluate the network on the test set. All results in the remainder of the paper are based on the test set.

3.2.2.3 Data post-processing

All blocks were processed independently by the neural network. To transform the predicted segmentation of the overlapping blocks back into a predicted segmentation of the entire plant, the overlap of the blocks had to be removed. This was done by again sorting the points per block. From the first block, the first 30,000 points were then selected. From the subsequent blocks, points 10,000-30,000 were selected and finally, from the last block the still missing points were selected. The number of points selected from the last block varies, because of the varying overlap between this block and the second to last block. By following this approach, for each point having multiple predictions, the one closest to the centre of a block was used.

3.2.3 Segmentation evaluation

To evaluate the performance of the point cloud segmentation, we used the intersection over union (IoU), which is a metric often used in the evaluation of 3D point cloud segmentation (Guo et al., 2020). The general formula for the IoU for class i is given in Equation 3.1. This value is computed for all classes and for each of the point clouds in the test set. The evaluation was based on all points for which the labelling was consistent, that is where the two manually assigned labels were identical.

$$\mathrm{IoU_i} = \frac{\mathrm{TP_i}}{\mathrm{TP_i} + \mathrm{FP_i} + \mathrm{FN_i}} \quad [-] \qquad \textit{Equation 3.1}$$

where TP$_i$ refers to the number of points correctly predicted to belong to class i (true positives), FP$_i$ refers to the number of points incorrectly predicted to belong to class i (false positives) and FN$_i$ refers to the number of points that belong to class i, but are predicted to belong to a different class (false negatives).

The average IoU-value can be computed either as a micro average or as a macro average of the IoU-values per class. The micro average, $\mathrm{IoU_{micro}}$, was calculated according to:

$$\mathrm{IoU_{micro}} = \frac{\sum_{i=1}^{k} \mathrm{TP_i}}{\sum_{i=1}^{k}(\mathrm{TP_i} + \mathrm{FP_i} + \mathrm{FN_i})} \quad [-] \qquad \textit{Equation 3.2}$$

where k is the number of classes. However, because of the class imbalance, this value is dominated by the leaf class. Therefore, we also included the macro average, IoU$_{macro}$, which is the average of the IoU over all classes:

$$\mathrm{IoU_{macro}} = \frac{\sum_{i=1}^{k} \mathrm{IoU_i}}{\mathrm{k}} \quad [-] \qquad \textit{Equation 3.3}$$

In the results of the paper, the IoU-values of the different segmentation approaches are compared to each other. The significance of the differences is tested using a one-sided Wilcoxon signed-rank test. The significance is reported as n.s. (not significant) when p>0.05, * when p<0.05, ** when p<0.01 and *** when p<0.001.

3.2.4 Experiments

In this section, we explain the three experiments that were performed to answer the research questions of this paper.

3.2.4.1 Spectral data

As introduced before, the laser scanners used to obtain the point cloud data capture spectral features in addition to the geometric data. The first research question of this paper is if the segmentation performance increases if the geometric data XYZ is enriched with spectral data RGB. To answer this research question, we have trained the network with and without the spectral data, as explained in section 3.2.2.2. In both cases the network was trained until the loss on the validation set did not decrease any further.

To get a deeper understanding of the effect of adding RGB data, we also compared the confusion matrices of the trained network with and without spectral data. The difference between the confusion matrix for XYZ and the confusion matrix for XYZRGB shows the change in errors made by the network when spectral data was added.

Finally, the cross-validation described in section 3.2.1.4 was done for this experiment. The results are shown in section 3.3.1.

3.2.4.2 Intra-observer variability

The second research question is what the intra-observer variability between annotated dataset A and annotated dataset B is and how it does affect the segmentation performance. The intra-observer variability is measured as the IoU between the two annotated datasets. For this experiment, a point is considered a TP if it has the same label in both annotated datasets. If this is not the case, the point counts as a FN for the label assigned in set A and as a FP for the label assigned in set B. If both annotated datasets are identical, this leads to IoU-values of 1 for each class, if there is no overlap the IoU is 0. The actually observed IoU-values are reported in section 3.3.2. We also present the confusion matrix, to give more insight in the differences between the two annotated datasets and how each class was labelled.

Additional to the results quantifying the intra-observer variability, we have trained the network first using annotated dataset A, then using annotated dataset B and finally using both annotated datasets together. In all these cases, we have used the combination of XYZ and RGB data as input for the segmentation method. The results are also shown in section 3.3.2.

3.2.4.3 Design of the phenotyping experiment

The final experiment was devoted to answer the research question how much the segmentation performance increases when the plant and its environment are simplified by optimising the phenotyping experiment. As explained in section 3.2.1.3, during the manual annotation, eight different classes were distinguished. However, the number of classes could be reduced by changing the way the crop maintenance was executed. For example, if the tendrils and ovaries were removed before scanning the plants, there would be no need for the network to learn how to recognise these parts. For the purpose of measuring plant architecture in a plant science or breeding setting, this is a feasible step to implement. Furthermore, the non-plant objects like the plant gutter, the pot and the supporting wire could be given a distinctive colour, such that they can be easily removed from the data in a pre-processing step. Finally, the node is an underrepresented class that is hard to label by hand. It goes beyond the scope of this paper to actually develop a method to do this, but it is reasonable to expect that if the stem and petiole are segmented, the node can also be detected at the intersection of these two. Based on these considerations, we adapted the original 8-class segmentation task to a 4-class segmentation task according to Table 3.2, by removing the ovary, tendril and non-plant points and relabelling the node points as stem points. The network was trained again using the 4-class segmentation task. In section 3.3.3, the obtained IoU-values and the confusion matrix are presented.

Table 3.2 – Classes used in the 8-class and the 4-class segmentation task. In the 4-class segmentation task, the classes ovary, tendril and non-plant were removed from the point cloud. The class node was merged with the class stem.

	Stem	**Node**	**Petiole**	**Leaf**	**Gr. point**	**Ovary**	**Tendril**	**Non-plant**
8-class	✓	✓	✓	✓	✓	✓	✓	✓
4-class	✓		✓	✓	✓			

3.3 Results

In this section, the results of the experiments defined in section 3.2.4 are presented. First, in section 3.3.1, the results of the segmentation based on geometric data are presented and compared to the segmentation based on the geometric data enriched with spectral data. In section 3.3.2, the intra-observer variability results of experiment 2 are shown. Finally, in section 3.3.3, the results of training the network on the 4-class segmentation task are shown. All results are based on the performance of the method on the test set of the data.

3.3.1 Spectral data

The IoU-values obtained when training the network on geometric data only (XYZ) and on the combination of geometric and spectral data (XYZRGB) are shown in Figure 3.6. The biggest improvement was found for the stem, for which the mean IoU increased from 0.41 to 0.70 ($p<0.001$). The IoU for the classes petiole, leaf and non-plant ($p<0.001$) as well as for the class tendril ($p<0.05$) also increased significantly. For the classes growing point, node and ovary, the mean IoU did increase, although not significantly. Overall, both the micro as well as the macro average of the IoU showed a significant improvement when spectral data was added.

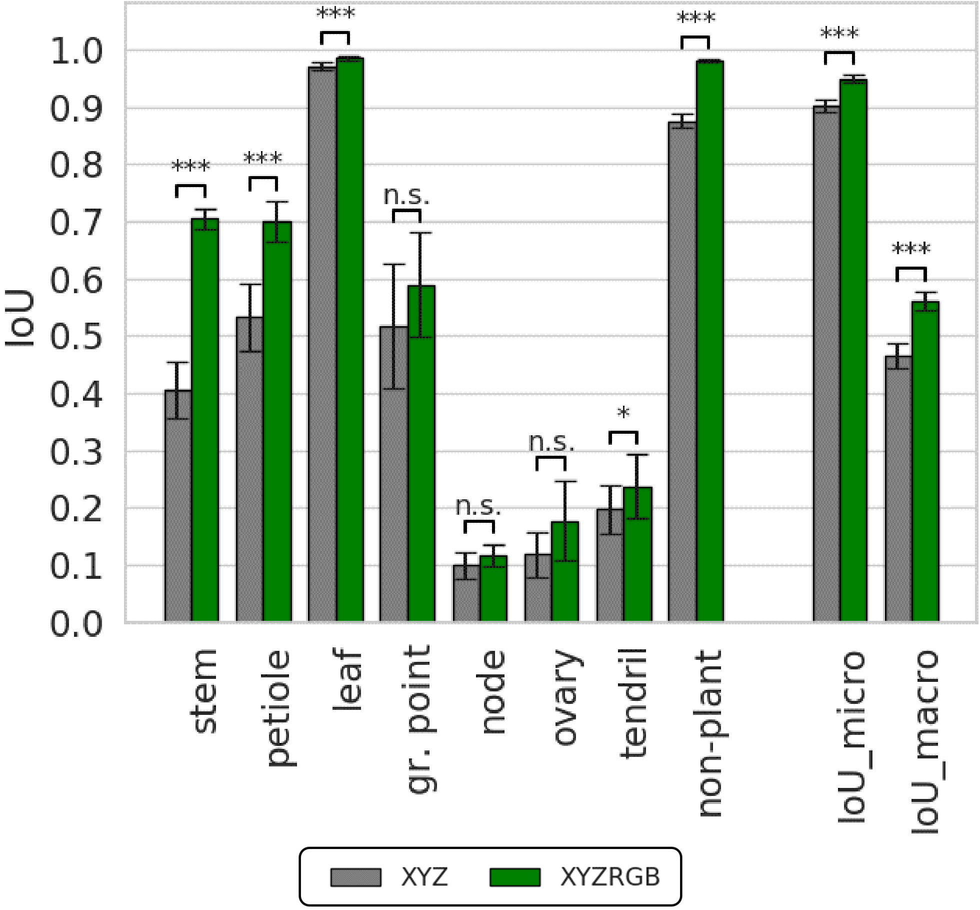


*Figure 3.6 - Performance on the test set based on only geometric data (XYZ, grey) and geometric and spectral data (XYZRGB, green). The bars give the mean IoU and the error bars show the 95% confidence interval on the mean. The asterisks indicate a significant improvement of the performance when adding spectral data (p<0.05 = *, p<0.01 = **, p<0.001 = ***).*

The confusion matrices for XYZ and XYZRGB are shown in Table 3.3, to give more insight in the prediction errors. Looking at the results for XYZ, it can be seen that 99.1% of the points manually labelled as leaf are also predicted to be leaf by the network. Also, for the non-plant objects, 94.0% of the points are correctly predicted. The most errors are made for the classes node, ovary and tendril. The node is mostly confused with the stem (37.9%) and the petiole (21.8%). For ovary, the errors are quite evenly distributed over stem, petiole, leaf and tendril. The points manually labelled as tendril are mostly confused with the stem and leaf.

In the confusion matrix of the network trained on XYZRGB, it can be seen that the percentage of correctly predicted points, shown on the diagonal, is higher than for the network trained on XYZ for all classes except for the class tendril. To visualise the change in performance, we subtracted the confusion matrix XYZ from the confusion matrix XYZRGB, see Table 3.4. Note that in this table, the sum of the rows equals 0%. In this table, if the performance of XYZRGB was better than XYZ (more correct predictions on the diagonal or less errors outside the diagonal), the cell is highlighted in green, if the performance of XYZRGB was worse than XYZ, the cell is highlighted in red.

The biggest improvement was observed for the stem, where the percentage of correct predictions increased by 34.1 percentage points, mainly because of less confusion with non-plant objects. For the other classes, except tendril, the percentage of correct predictions also increased when the point clouds were enriched with spectral data. However, also some values outside of the diagonal increased, indicating an increased error rate. The highest value (24.2%) was found for points that were manually labelled as node and that were predicted as stem. Apparently, the node points that are, due to the spectral data, no longer incorrectly predicted to be petiole (-9.4%), leaf (-7.3%) or non-plant (-15.3%), are now (partly) incorrectly predicted to be stem. Still, also the percentage of correct predictions for the class node increased.

Table 3.3 – Confusion matrices for XYZ (above) and XYZRGB (below). The values on the diagonal (marked in bold) show the true positive rate. Each row in these matrices corresponds to the points given a specific class in the manual annotation and shows the division of these points over the available classes as predicted by the network. The values sum up to 100% for every row (exceptions due to rounding of the numbers). The correct predictions are highlighted in green, the wrong predictions are highlighted in red, brighter colours indicate higher values.

			Predictions							
			Stem	Petiole	Leaf	Gr. point	Node	Ovary	Tendril	Non-plant
Labels	Stem	XYZ	**54.7**	6.3	8.5	1.1	3.3	1.1	2.4	22.6
	Petiole		4.6	**72.0**	17.0	0.7	1.7	2.5	0.5	1.0
	Leaf		0.1	0.2	**99.1**	0.2	0.0	0.0	0.2	0.2
	Gr. point		0.1	0.0	9.9	**76.7**	0.0	0.0	0.3	13.0
	Node		37.9	21.8	7.7	1.5	**11.5**	3.1	0.4	16.1
	Ovary		20.0	23.1	17.8	0.0	5.2	**9.8**	21.0	3.2
	Tendril		23.1	1.6	32.0	2.1	0.7	1.4	**33.2**	5.9
	Non-plant		3.3	0.5	1.8	0.3	0.1	0.0	0.1	**94.0**

			Predictions							
			Stem	Petiole	Leaf	Gr. point	Node	Ovary	Tendril	Non-plant
Labels	Stem	XYZRGB	**88.8**	3.2	0.6	1.1	4.6	0.5	0.0	1.2
	Petiole		6.1	**86.3**	3.7	0.6	2.4	0.7	0.3	0.1
	Leaf		0.1	0.2	**99.4**	0.3	0.0	0.0	0.0	0.0
	Gr. point		3.0	0.3	14.1	**82.2**	0.0	0.0	0.0	0.4
	Node		62.0	12.4	0.4	1.4	**20.4**	2.2	0.3	0.8
	Ovary		15.8	29.4	0.7	0.0	7.6	**32.6**	12.7	1.2
	Tendril		22.1	1.7	33.6	3.1	1	2.5	**31.1**	5.0
	Non-plant		0.8	0.0	0.5	0.1	0.0	0.0	0.0	**98.5**

Table 3.4 – Difference between confusion matrix XYZRGB and confusion matrix XYZ. Green values indicate less confusion (more correct predictions on the diagonal or less errors outside the diagonal) and red values indicate more confusion. The sum of percentages in the rows adds up to 0% (exceptions due to rounding of the numbers).

			Predictions							
			Stem	Petiole	Leaf	Gr. point	Node	Ovary	Tendril	Non-plant
Labels	Stem	XYZRGB - XYZ	**34.1**	-3.1	-7.9	0.0	1.3	-0.6	-2.4	-21.4
	Petiole		1.5	**14.2**	-13.3	-0.2	0.7	-1.9	-0.2	-0.9
	Leaf		0.0	0.0	**0.3**	0.0	0.0	0.0	-0.2	-0.2
	Gr. point		2.9	0.3	4.1	**5.5**	0.0	0.0	-0.3	-12.6
	Node		24.2	-9.4	-7.3	0.0	**8.9**	-0.9	-0.1	-15.3
	Ovary		-4.2	6.3	-17.1	-0.01	2.4	**22.9**	-8.3	-2.0
	Tendril		-1.0	0.1	1.6	1.0	0.3	1.1	**-2.1**	-0.9
	Non-plant		-2.5	-0.5	-1.3	-0.2	0.0	0.0	-0.1	**4.5**

Cross-validation

The range between minimum and maximum IoU-value for the four cross-validation runs is shown in Figure 3.7. If this range is small, it means there is a low effect on the segmentation performance for the different cross-validation runs. On the other side, a large range means that the specific composition of the training, validation and test set has an effect on the segmentation performance. The highest difference between maximum and minimum IoU over the four cross-validation runs was observed for the class tendril, including spectral data. Here, the maximum IoU observed was 0.22 higher than the minimum IoU. The most consistent performance was observed for the class leaf, also when spectral data was included. In this case, the highest IoU was only 0.01 higher than the lowest IoU.

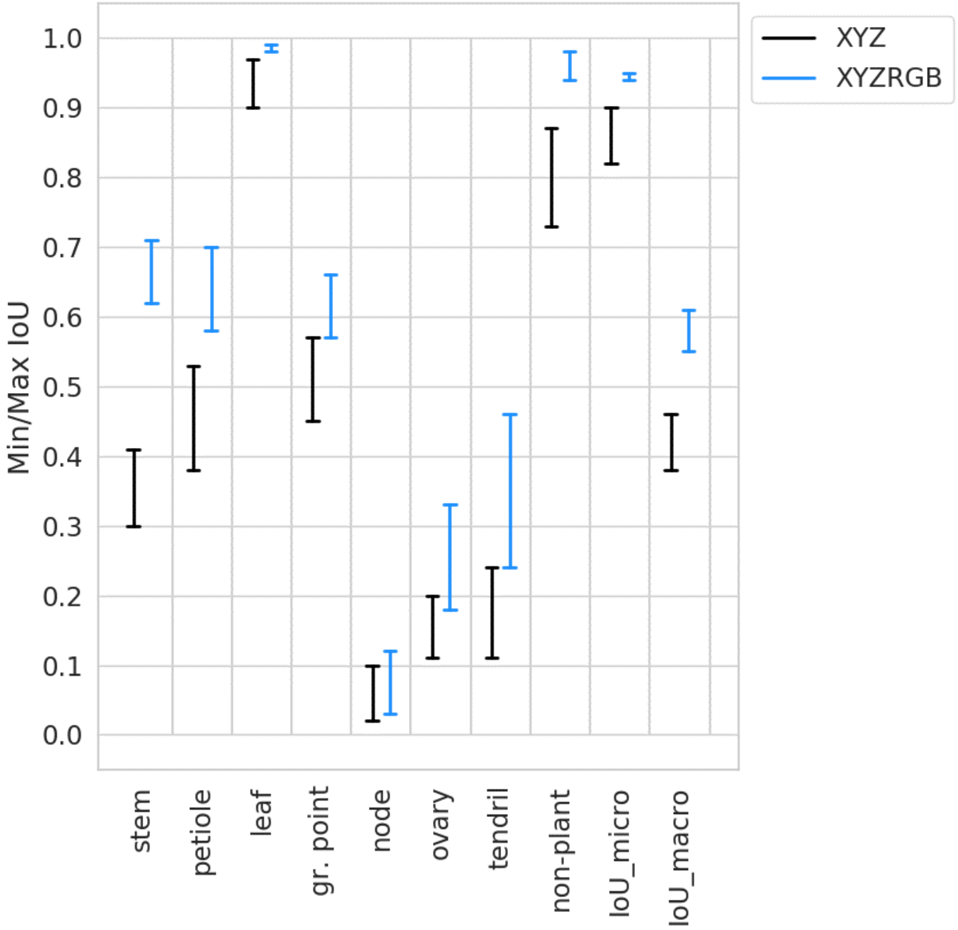


Figure 3.7 – Minimum and maximum IoU-values observed over the four cross-validation runs for XYZ (black lines) and XYZRGB (blue lines).

To verify if the observed variation influences the results of experiment 1, we tested if the mean IoU-values obtained for XYZRGB were higher than for XYZ, for all four cross-validation runs, using the Wilcoxon signed-rank test. The significance of the performed tests is reported in Table 3.5. Although there are some differences for growing point, node, and ovary, these do not change the conclusions of this experiment: adding spectral data significantly improves the segmentation of stem, petiole, leaf, tendril and non-plant objects, as well as the overall segmentation quality as measured by the $\text{IoU}_{\text{micro}}$ and $\text{IoU}_{\text{macro}}$.

*Table 3.5 – Results of testing whether the IoU for the combination of XYZRGB was significantly higher than the IoU for XYZ only for the four cross-validation runs (Not significant = n.s., p<0.05 = *, p<0.01 = **, p<0.001 = ***). Note that the first column shows the same significance levels as shown in Figure 3.6.*

	CV1	CV2	CV3	CV4
Stem	***	***	***	***
Petiole	***	***	***	***
Leaf	***	***	***	***
Gr. point	n.s.	**	n.s.	n.s.
Node	n.s.	n.s.	n.s.	n.s.
Ovary	n.s.	*	*	n.s.
Tendril	*	***	***	***
Non-plant	***	***	***	***
$\text{IoU}_{\text{micro}}$	***	***	***	***
$\text{IoU}_{\text{macro}}$	***	***	***	***

3.3.2 Intra-observer variability

As explained in section 3.2.4.2, the intra-observer variability was measured as the IoU between annotation A and annotation B. The results are shown in Figure 3.8. It can be seen that mainly for the classes leaf and non-plant, the correspondence between the two annotations is very high (mean IoU of 0.99 and 0.98 respectively). The mean IoU for stem (0.86) and petiole (0.85) is also high. For the classes growing point (0.67), tendril (0.74) and ovary (0.55) the mean IoU value is lower. The lowest value is found for the class node, with a mean IoU of 0.49. The micro average of the IoU-values is 0.98 and the macro average of the IoU-values is 0.77. The micro average is higher than the macro average, because the majority of points belongs to the classes leaf and non-plant, which both have a very high mean IoU-value.

Lower IoU-values in this context mean that more points in that class were labelled differently in the two annotation sets. Apparently, there is uncertainty about the correct label for these points, even if labelled by the same annotator. The 95% confidence interval shown by the error bars is small, indicating that the intra-observer variability is stable over the different point clouds.

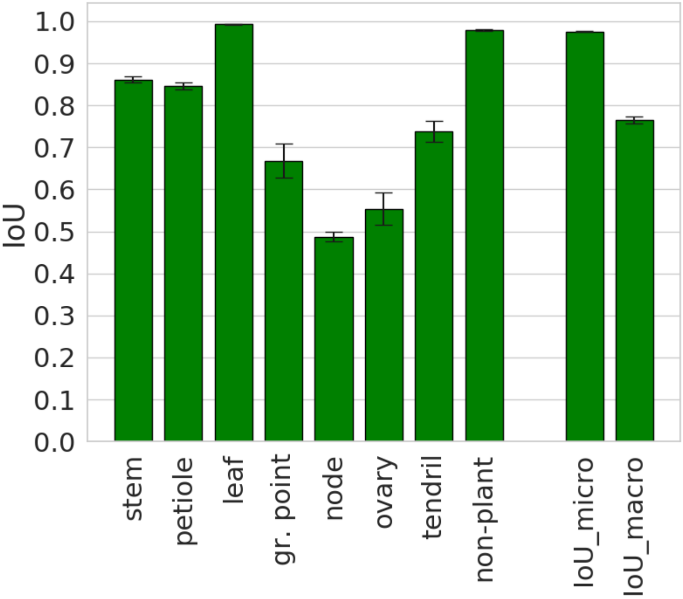


Figure 3.8 – Mean IoU between annotation A and annotation B. The bars indicate the mean IoU and the error bars show the 95% confidence interval on the mean. Note that a high IoU-value corresponds with a low intra-observer variability while a low IoU-value corresponds with a high intra-observer variability.

To give more insight in the confusion between classes, the confusion matrix between annotation A and annotation B is shown in Table 3.6. Also here, the classes with the highest agreement between annotation A and annotation B are leaf and non-plant. The lowest agreement between annotation A and B (65.6 %) is for the class node. For the points labelled as node in annotation A, 15.3 % was labelled as stem and 10.7 % was labelled as petiole in annotation B. Also note that 13.2% of the points labelled as growing point in annotation A, was labelled as leaf in annotation B.

Table 3.6 – Confusion matrix annotation A versus annotation B. Values on the diagonal (annotation B equals annotation A) are marked in bold. Values outside the diagonal that stand out in deviation (indicating disagreement between annotation A and B) are highlighted in red. Each row corresponds to the points given a specific class in annotation A and shows the division of these points over the available classes as labelled in annotation B. The sum of percentages in the rows adds up to 100%.

		Annotation B							
		Stem	Petiole	Leaf	Gr. point	Node	Ovary	Tendril	Non-plant
Annotation A	Stem	**92.0**	1.1	0.8	0.7	3.2	0.5	0.3	1.5
	Petiole	1.6	**91.4**	1.6	0.4	2.6	1.5	0.6	0.4
	Leaf	0.0	0.0	**99.7**	0.2	0.0	0.0	0.0	0.1
	Gr. point	0.6	0.3	13.2	**83.7**	0.2	0.2	0.6	1.3
	Node	15.3	10.7	1.5	0.5	**65.6**	3.0	1.4	2.2
	Ovary	2.8	3.3	1.7	0.5	1.8	**87.0**	0.5	2.4
	Tendril	0.8	0.9	5.2	0.7	0.6	0.5	**87.7**	3.6
	Non-plant	0.2	0.0	0.3	0.1	0.0	0.1	0.2	**99.2**

Training on annotation A, B, or the combination of A and B

The performance of the network when trained using only annotation A, only annotation B or the combination of annotation A and B is shown in Figure 3.9. There is no observable difference between training on annotation A or annotation B, indicating that the quality of the two annotation sets is similar. When the network is trained on both annotation sets, there is a slight improvement in performance. In some cases, this improvement is significant. This shows that there is additional information in the second annotation set, from which the network can learn.

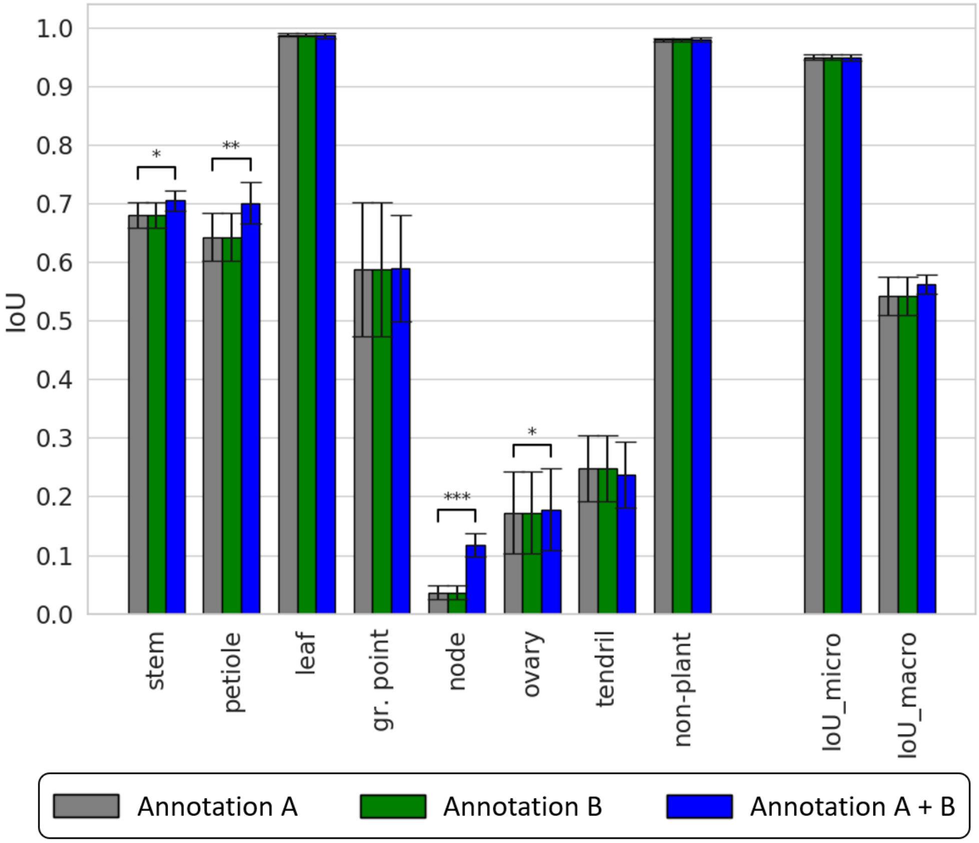


*Figure 3.9 - IoU for the test set based on training with only annotation A or B and training based on annotation A and B. The bars give the mean IoU and the error bars show the 95% confidence interval on the mean. The asterisks indicate a significant improvement of the performance when trained on two annotation sets as compared to the network trained on one of the datasets (p<0.05 = *, p<0.01 = **, p<0.001 = ***).*

3.3.3 Design of the phenotyping experiment

In experiment 3, we trained the network for a 4-class (stem, petiole, growing point and leaf) segmentation and compared it to the previously used 8-class segmentation. The results are shown in Figure 3.10. For all classes, the mean IoU is significantly higher for the 4-class segmentation approach. For the 4-class segmentation as compared to the 8-class segmentation, the mean IoU for the stem increased from 0.70 to 0.90, for the petiole from 0.70 to 0.83 and for the growing point from 0.59 to 0.73. For the leaf, which already had a high IoU, a significant but very small increase of the IoU was observed. The overall increased performance is also reflected in the $\mathrm{IoU}_{\mathrm{micro}}$, rising from 0.95 to 0.98 and the $\mathrm{IoU}_{\mathrm{macro}}$, rising from 0.74 to 0.86.

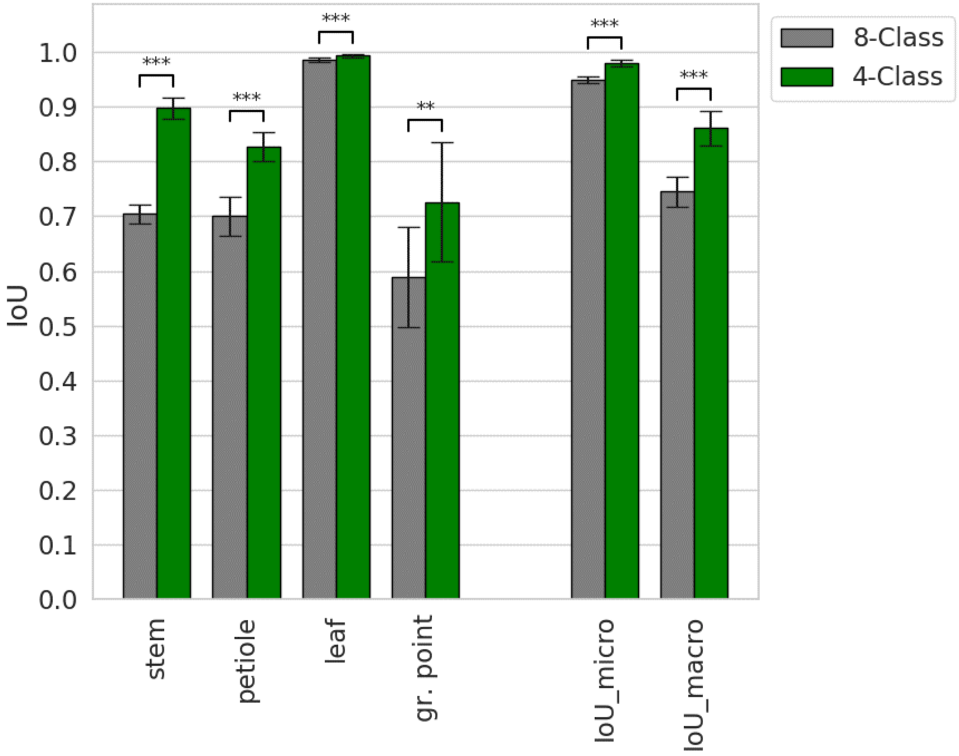


*Figure 3.10 – Performance on the test set for the original 8-class and the simplified 4-class segmentation task. The bars give the mean IoU for the test set and the error bars show the 95% confidence interval on the mean. The asterisks indicate a significant improvement of the performance for the 4-class segmentation as compared to the 8-class segmentation (p<0.05 = *, p<0.01 = **, p<0.001 = ***).*

The confusion matrix for the 4-class segmentation is shown in Table 3.7 (for the confusion matrix of the 8-class segmentation, see Table 3.3). All values on the diagonal are higher for the 4-class segmentation than for the 8-class segmentation. The errors made by the 4-class segmentation are similar to the 8-class segmentation. For the stem, the false negatives are mainly caused by confusion with the petiole (3.1 %), while the false positives are caused by petiole (6.9 %) and growing point (2.1 %). The petiole has most false negatives when wrongly classified as stem (6.9 %) or leaf (2.7 %). The growing point does not have many false positives. False negatives for growing point are mainly caused by stem (2.1%) and leaf (2.4%).

Table 3.7 – Confusion matrices for the 4-class segmentation. Values on the diagonal (correct predictions) are marked in bold. Values outside the diagonal that stand out in deviation are highlighted in red.

		Predictions			
		Stem	Petiole	Leaf	Gr. point
Labels	Stem	**96.7**	3.1	0.1	0.1
	Petiole	6.9	**90.3**	2.7	0.2
	Leaf	0.1	0.1	**99.5**	0.3
	Gr. point	2.1	1.3	2.4	**94.2**

3.4 Discussion & recommendations

In this paper, we have demonstrated the ability of a PointNet++ -based method to segment partially complete point clouds of cucumber plants. The results clearly show that the segmentation improved when spectral data was added to the point clouds. Although there were slight differences between the four cross-validation runs, all four show significant improvements for stem, petiole, leaf, tendril and non-plant objects. For growing point, node and ovary, the mean IoU also increased, although not significantly.

The difference between the confusion matrices for XYZ and XYZRGB (see Table 3.4) shows the effect of adding RGB data on the confusion of the network. The main contribution of spectral data was to reduce confusion of parts that have a distinct colour. For example, the improved segmentation of the stem comes mainly from less confusion with non-plant parts. The supporting wire to which the plants were attached was red, while the stem was green, which explains the added value of RGB data to separate the two classes. However, being able to better segment this supporting wire, resulted in a higher confusion between stem and node. This can be explained, because while the supporting wire has a distinct colour, the stem and the node have very similar colours. The class ovary also showed a relatively large benefit from the spectral data, because this class contained the yellow flowers of the otherwise mostly green cucumber plant.

The added value of spectral data as shown in this paper was based on RGB data in the visible part of the spectrum. In order to improve the segmentation of plant parts that have similar colours, like the node and stem, other parts of the spectrum like ultraviolet, near infrared or shortwave infrared could be relevant (Brugger et al., 2019; Kamilaris & Prenafeta-Boldú, 2018; Roitsch et al., 2019). Enriching the geometric data with spectral data in these parts of the spectrum could further improve the segmentation results. Future research should demonstrate which parts of the spectrum are most relevant.

The results obtained in this research agree with the research of Turgut et al. (2020), in which point clouds of rosebushes (XYZ only) were segmented into flower, leaf and stem. Using PointNet++, the best IoU-values reported in their research are 0.77 for stem, 0.95 for leaf and 0.73 for flower. When synthetic data was used for pre-training, these values increased to 0.83 for stem, 0.96 for leaf and 0.79 for flower. In our case, the class flower was not present, as flowers and emerging fruits were both labelled as ovary. Our results showed an IoU of 0.70 for stem, 0.99 for leaf and 0.18 for ovary. Of course, the architecture of roses and cucumber plants is not the same, but it is interesting to see that for leaf and stem similar performances were achieved. In our case, only 0.54 % of the points were labelled as ovary, where in the rose segmentation of Turgut et al., 4.81 % of the points in the training set was labelled as flower. This could explain why the IoU for ovary in our research was lower than the IoU for flower as observed by Turgut et al.

Class imbalance is an issue in supervised learning and as expected classes for which more examples were present in the training data achieved higher IoU values than classes with fewer examples in the training data. Although we expect that the difference in segmentation performance between over- and underrepresented classes does not change the answers to the questions addressed in this paper, improving the segmentation of underrepresented classes is highly relevant for the plant phenotyping domain in general. Of course, this depends on the specific traits of interest in an experiment. For example, when measuring internode length or flower and fruit development, the current segmentation performance is probably not sufficient. To improve the segmentation of underrepresented classes, we suggest to follow the approach of Turgut et al. (2020) and use synthetic data to obtain a more balanced dataset. Another approach could be to use data augmentation methods focused on the underrepresented classes. For example, regions of the plant containing more of the underrepresented classes could be augmented and repeated more often in the training procedure as compared to overrepresented classes. Another suggestion is to combine all underrepresented classes and segment them from the overrepresented classes in a first segmentation step. A second network can then be trained to segment the underrepresented classes.

Having a multi-step approach also brings another advantage. In the pre-processing of the point clouds, we applied a voxel grid filter. This was done because of the high point density in mainly the leaves. Without this filter, a leaf could easily contain more than 40,000 points, meaning that blocks would be formed containing only a part of a leaf. The voxel grid filter prevented this from happening by down sampling the point cloud. However, since at that time the segmentation was not yet known, the filter was also applied on the other classes and thus also removed points of the classes that were difficult to segment. If a multi-step approach is implemented, in between the steps the resolution of the point cloud could be increased again. For example, in our case, the first step would result in a filtered point cloud where the leaves and the non-plant objects are removed. The remaining points can then be mapped back in the unfiltered point cloud, and a nearest neighbour algorithm can select points to increase the resolution. We plan to test this is in future work.

Since the proposed method heavily leans on the ground-truth data, inter- and intra-observer variability might have a serious impact on the results. The results presented in section 3.3.2 show that the IoU between the two annotation sets of our research varies per class. The most consistently labelled classes were leaf and non-plant, having an IoU of 0.99 and 0.98. The lowest IoU-values were observed for ovary and node at 0.55 and 0.49. Leaves and non-plant objects could be easier to recognise for labellers, but these parts are also larger in size than for example node and ovary. For larger objects, there are relatively fewer boundary points. As the disagreement between annotation set A and B mainly occurs at the boundary from one class to another, this disagreement has a bigger effect on small objects like the node and ovary.

The segmentation results showed similarities with the intra-observer variability. For the classes with high intra-observer variability or a low IoU between the two annotated sets (node, ovary, tendril, and growing point), the proposed method showed a low segmentation performance, except for the growing point, which shows that these classes are challenging to segment for the machine as well as for the human. Similarly, the classes with low intra-observer variability (leaf and non-plant) were also segmented very well.

Overall, the results of this experiment indicated that consistent point cloud annotation is a challenging task. The improved segmentation performance observed for a network trained on the combination of annotation set A and B, as compared to a network trained on only A or B, indicates that taking label uncertainty into account can improve the quality of the segmentation. Future work should investigate how to improve the collection of proper ground truth data to learn from and how variation between observers could be used in the training procedure.

In the final experiment, we reduced the number of classes in the segmentation task. Although for this experiment, this was done by simulating the data, in future experiments this could be achieved by using a distinctive colour for non-plant objects to allow automatic removal of these parts and by changing the crop maintenance such that plant parts not relevant for the plant architecture are not in the data. For dedicated breeding trials this is a feasible option. However, if plants of a commercial grower are measured, there are of course limitations.

The results presented in this paper are based on point clouds obtained using PlantEye F500 laser scanners. These scanners have a high spatial resolution and provide coloured point clouds. Using other data acquisition methods could lead to differences in for example point density and completeness of the data. Also, other sensors might be sensitive to changes in illumination conditions. Changing the data acquisition as well as changing the environment in which the plants grow, might cause the need for retraining of the network. Even after retraining, differences in the quality of the data can lead to differences in the performance of the network. Still, we expect that the conclusions of this paper remain valid, as the objective of this paper was not to reach the highest possible IoU, but rather to show the added value of spectral data for plant-part segmentation.

3.5 Conclusion

The results of this paper demonstrate the ability of a PointNet++ -based method to segment partially complete point clouds of cucumber plants. It was shown that the availability of spectral data significantly improves the segmentation of stem, petiole, leaf, tendril and non-plant objects, as well as the overall segmentation quality as measured by the $\mathrm{IoU}_{\mathrm{micro}}$ and $\mathrm{IoU}_{\mathrm{macro}}$. The $\mathrm{IoU}_{\mathrm{micro}}$ increased from 0.90 to 0.95 and the $\mathrm{IoU}_{\mathrm{macro}}$ increased from 0.46 to 0.56 when spectral data was added. The biggest improvement was observed for the stem, where the IoU rose from 0.41 to 0.70. The intra-observer variability showed differences per class and showed that consistent annotation of point clouds is a challenging task. The lowest agreement between the two hand-labelled datasets was observed for the node, having an IoU of 0.49. Combining the two labelled datasets resulted in a small but significant improvement of the IoU for stem, petiole, node and ovary. Finally, we showed that a careful design of the plant phenotyping experiment can improve the segmentation quality. For all classes, the IoU increased when the segmentation task contained fewer classes. A practical example to achieve this would be to provide a distinctive colour for non-plant objects and to apply crop maintenance specific for the experiment, removing irrelevant plant parts on forehand.

This paper focuses on the segmentation of point clouds. However, to obtain phenotypic data sets, segmentation only is not sufficient. Additional methods to obtain phenotypic measurements from the segmented point clouds are needed. This paper contributes to the understanding and optimisation of the segmentation process and with that it supports the development of these follow-up methods. This brings the overall goal to develop automated phenotyping methods for complex plant traits one step closer.

3.6 Appendix A

Table 3.A1 – The number of blocks per scan (left (L) and right (R) of the plant gutter) for the 11 measurement days. The last column shows the average number of blocks per scan for the 11 days. Note that if the two annotated datasets are used, the number of blocks doubles.

Plant	A1		A2		A3		A4		A5		A6	
Side of gutter	L	R	L	R	L	R	L	R	L	R	L	R
Measurement day:												
1	2	3	2	2	3	3	3	3	3	3	3	3
2	3	3	2	3	4	4	3	4	4	4	3	3
3	3	4	3	3	4	5	4	5	4	5	4	4
4	4	4	3	4	5	5	5	5	4	5	4	4
5	5	5	4	5	5	6	6	6	5	5	5	5
6	5	6	4	5	6	7	6	6	5	6	5	5
7	6	7	5	6	7	7	7	7	6	6	6	6
8	7	8	6	6	8	8	8	8	7	7	6	7
9	8	9	7	7	8	8	9	8	7	7	7	7
10	9	9	7	7	8	9	9	9	8	8	7	8
11	9	10	8	8	8	9	9	9	8	9	8	8

Plant	B1		B2		B3		B4		B5		B6		Average
Side of gutter	L	R	L	R	L	R	L	R	L	R	L	R	A1-B6
Measurement day:													
1	3	3	3	3	4	3	3	3	3	3	3	3	2.9
2	4	4	4	4	5	4	4	4	4	3	2	3	3.5
3	5	4	4	5	5	5	5	4	4	4	3	3	4.1
4	5	5	5	5	6	5	5	5	5	5	4	4	4.6
5	6	6	5	5	6	6	5	6	6	5	4	4	5.3
6	6	6	6	5	7	6	6	6	6	6	5	5	5.7
7	7	7	6	6	7	7	6	7	7	7	5	6	6.4
8	7	7	7	7	8	8	8	8	7	7	6	6	7.2
9	8	8	7	8	8	8	9	8	8	8	7	6	7.7
10	8	9	9	9	9	9	9	9	8	9	7	7	8.3
11	9	10	10	10	10	10	10	10	9	10	8	8	9.0

Chapter 4

Improved point-cloud segmentation for plant phenotyping through class-dependent sampling of training data to battle class imbalance

This chapter has been published as:

Boogaard, F. P., van Henten, E. J., & Kootstra, G. (2022). Improved Point-Cloud Segmentation for Plant Phenotyping Through Class-Dependent Sampling of Training Data to Battle Class Imbalance. *Frontiers in Plant Science*, *13:838190*. https://doi.org/10.3389/fpls.2022.838190

Nomenclature

Point features	r, g, b	Spectral features: red, green, blue
	x, y, z	Geometric features (x, y, z location of point)
Division into chunks	a	Total number of anchor points per point cloud
	a_c	Number of anchor points per class c, per point cloud
	n	Total number of points in point cloud
	n_c	Total number of points in class c
	k	Number of points per chunk
Shannon Entropy	d	Dataset, refers to the manually labelled dataset, or one of the training, validation and test sets
	C	Number of classes (C = 8 in the current research)
	$h(d)$	Shannon entropy for dataset d
	p_c^d	Probability that a randomly chosen point belongs to class c in dataset d
Evaluation	IoU_c, $\mathrm{IoU}_{\mathrm{micro}}$, $\mathrm{IoU}_{\mathrm{macro}}$	Intersection-over-Union for class c, or as micro or macro average
	$\mathrm{TP}, \mathrm{FP}, \mathrm{FN}$	True positive, False positive, False negative
	P, R	Precision, Recall

Abstract

Plant scientists and breeders require high-quality phenotypic data. However, obtaining accurate manual measurements for large plant populations is often infeasible, due to the high labour requirement involved. This is especially the case for more complex plant traits, like the traits defining the plant architecture. Computer-vision methods can help in solving this bottleneck. The current work focuses on methods using 3D point cloud data to obtain phenotypic datasets of traits related to the plant architecture. A first step is the segmentation of the point clouds into plant organs. One of the issues in point-cloud segmentation is that not all plant parts are equally represented in the data and that the segmentation performance is typically lower for minority classes than for majority classes.

To address this class-imbalance problem, we used a common practice to divide large point clouds into chunks, that were independently segmented and recombined later. In our case, the chunks were created by selecting anchor points and combining those with points in their neighbourhood. As a baseline, the anchor points were selected in a class-independent way, representing the class distribution in the original data. Then, we propose a class-dependent sampling strategy to battle class imbalance. The difference in segmentation performance between the class-independent and the class-dependent training set was analysed first. Additionally, the effect of the number of points selected as the neighbourhood was investigated. Smaller neighbourhoods resulted in a higher level of class balance, but also in a loss of context that was contained in the points around the anchor point.

The overall segmentation quality, measured as the mean intersection-over-union (IoU), increased from 0.94 to 0.96 when the class-dependent training set was used. The biggest class improvement was found for the ‘node’, for which the percentage of correctly segmented points increased by 46.0 percentage points. The results of the second experiment clearly showed that higher levels of class balance did not necessarily lead to better segmentation performance. Instead, the optimal neighbourhood size differed per class. In conclusion, it was demonstrated that our class-dependent sampling strategy led to an improved point-cloud segmentation method for plant phenotyping.

4.1 Introduction

Plant scientists and breeders require high-quality phenotypic data. For example, phenotypic measurements of plant architecture-related traits are relevant to study the genotype-phenotype-environment relationship in the light of plant architecture. The plant architecture is the set of traits defining the three-dimensional (3D) organisation of the plant parts (Reinhardt & Kuhlemeier, 2002) and is an important indicator of plant stress (Fahlgren et al., 2015; Paulus, 2019; Suter & Widmer, 2013). Traits related to plant architecture are for example internode length, leaf angle, and leaf area. Measuring these traits manually with a high accuracy and a high temporal resolution is infeasible, due to the high labour requirement involved. Therefore, the availability of high-quality phenotypic data is often limited, especially for more complex traits, such as the traits related to plant architecture.

Computer-vision methods can increase the accuracy of the trait measurements and reduce the amount of manual labour involved. This allows to scale the phenotypic dataset in the number of time points per measurement as well as in the number of plants measured. Because of the complex 3D organization of the plant parts, we focused on methods that are based on 3D point clouds using deep neural networks. In these methods, a first step that is often used, is to semantically segment the point cloud, such that for each point it is known to which plant part it belongs. The segmented point cloud can then be used as the basis for measuring plant traits. The aim of our work is to develop methods that can help in measuring traits related to the plant architecture to increase the availability of high-quality phenotypic data.

An issue that was identified in previous work on point-cloud segmentation, is that point clouds of plants typically have a high level of class imbalance, with, for instance, an abundance of leaf points, but only few points belonging to classes such as ‘flower’ or ‘node’ (Boogaard et al., 2021; Turgut et al., 2022). This has consequences for semantic- segmentation methods, with the segmentation quality for the underrepresented classes lagging behind that of the overrepresented classes. For example, in the point clouds used by Boogaard et al. (2021), each of the classes ‘node’, ‘ovary’, and ‘tendril consisted of less than 1% of the points. The highest Intersection-over-Union (IoU, a metric to measure segmentation quality) reported for these classes was 0.23, while for the majority class ‘leaf’, the IoU was 0.99. In point clouds of roses (Turgut et al., 2022), the ‘stem’ and ‘flower’ were underrepresented as compared to the ‘leaf’. The highest observed IoU-values for these underrepresented classes were 0.77 (‘stem’) and 0.73 (flower), while the IoU for the overrepresented ‘leaf’ was 0.95. In the current work, we propose a method to improve the segmentation of underrepresented classes, based on the level of class balance in the training data for the neural network.

4.1.1 Class imbalance – a literature review

The training procedure of a neural network consists of two main steps, creating the training data and training the network. In the first step, the available data needs to be manually labelled and transformed into a training, validation, and test set. The training data is then used to train the network. The second step, training, depends on a set of hyper parameters like the neural network architecture, the loss function, and the possibility of using pre-trained weights. In both steps, methods have been presented in literature to deal with class imbalance. We first discuss the approaches dealing with the training data and then the approaches dealing with the network and training procedure, in both cases focusing on 3D point clouds.

When preparing the training data, it is possible to repeat samples of underrepresented classes more often than samples of overrepresented classes. In the work of Lin and Nguyen (2020), randomly selected points of underrepresented classes were duplicated and randomly selected points of overrepresented classes were removed. This approach changes local point densities and structures, which could be prevented by oversampling entire objects instead of individual points. However, it is not straightforward to do this in semantic segmentation, since the recognition of an object also depends on the context in which it is presented. For example, the difference between a petiole and a stem is mainly visible because of the surrounding plant structure. One approach to do oversampling in a semantic-segmentation task was presented for the segmentation of LiDAR scans of urban areas. The point clouds in this work were divided into subsets, in this work referred to as *chunks*. A chunk was then duplicated if more than 70% of the points in that chunk belonged to underrepresented classes (Poliyapram et al., 2019). The possibility of applying data augmentation to duplicated chunks was added by Griffiths and Boehm (2019b), again in the context of 3D scanning and segmentation of an urban environment.

An alternative approach to reduce the level of class imbalance in the training set is to use synthetic data. An example was published by Griffiths and Boehm (2019a). They provided a large synthetic dataset of an urban environment in which the presence of small structures like poles and street furniture was increased, as compared to, for example, roads and pavements. In the plant domain, a synthetic dataset of sweet-pepper images including class and depth labels was generated by Barth et al. (2018) and in the work of Turgut et al. (2022), synthetic 3D rosebush models were used to train neural networks to segment plants from the ROSE-X dataset (Dutagaci et al., 2020). Such synthetic datasets could be modified to increase the focus on minority classes.

Also during training, class imbalance can be addressed. The training of a neural network is based on the loss, which is calculated from the difference between a predicted class and the ground truth. A common loss function for semantic segmentation is the cross-entropy loss, which gives equal weights to errors in each of the classes. The focus of a learning algorithm can be shifted towards minority classes by adding per-class weights to the loss function. These weights are then used to increase the loss for minority classes and decrease the loss for majority classes. This approach is known as weighted cross-entropy loss (Milioto et al., 2019). Focal loss is another variant of a weighted loss function in which the loss is inversely scaled by the probability that a point is of a certain class, which allows to obtain focus on minority classes (T. Lin et al., 2017).

Another way to deal with class imbalance is through pre-training of a neural network. Pre-training means that the network is first trained on a larger, often publicly available dataset, to learn common features. The dataset-specific features are then learnt in a second training procedure, based on the specific training set. This approach was adjusted to learn features of underrepresented classes first, before moving to the overrepresented classes, known as incremental transfer learning (Sander, 2020). The idea was to limit the dataset first to the underrepresented classes meaning that it is easier for the network to learn features of these classes. Then, the overrepresented classes are added, possibly in multiple steps. A strict stopping condition is required, to prevent overfitting when the overrepresented classes have been added.

Although all the above methods alleviate the problem of class imbalance to some extent, new methods are needed to further improve the segmentation of underrepresented classes. In the current work, we focus on improving the level of class balance in the training set. The methods dealing with the network and training procedure are out of the scope of the current work.

4.1.2 Contributions of the paper

In the literature presented above, a common approach was to divide the point clouds into chunks that were then independently processed by the network. The main reason to create these chunks was that large point clouds cannot be processed at once for reasons of limited available memory storage and computational power. However, the chunks were either created based on fixed volumes, or by applying a sliding window that covered the entire point cloud.

Instead of covering the entire point cloud, our aim was to show that the procedure to create the chunks could be designed such that the chunks were directly created with a focus on minority classes to improve the class balance in the training set. The main principle of our proposed method was based on two steps. First, anchor points were sampled from the point cloud and second, a set of points around each anchor point was selected as the chunk.

The focus of our method was to increase the level of class balance in the training set, based on the hypothesis that a higher level of class balance would lead to a better segmentation method. To test this, first, a reference training set was created in a class-independent way. In this class-independent training set, the class distribution of the sampled anchor points was proportional to the original class distribution. A second, class-dependent, training set was then generated, in which the class distribution of the sampled anchor points was inversely proportional to the original class distribution. In the first experiment, the added value of the class-dependent sampling approach was tested by using both training sets to train a neural network and comparing the performance on the segmentation task.

Although the anchor points were sampled from a specific class, the neighbourhood could contain points of different classes. So, even if an equal amount of anchor points was selected for each class, the neighbouring points of different classes reduced the level of class balance in the training set. By decreasing the number of points in the selected neighbourhood, this effect could be reduced. However, the neighbourhood of an anchor point does contain relevant information about the context of the anchor point. The second experiment, therefore, focused on the effect of the number of points added as the neighbourhood on the level of class imbalance and the segmentation performance.

In this work, cucumber was used as a model crop. The neural network architecture used for the segmentation was PointNet++ (Qi et al., 2017). PointNet++ is one of the top performing neural network architectures for point cloud segmentation (Guo et al., 2020). In the plant domain, PointNet++ was for example used in roses (Turgut et al., 2022) and in cucumber (Boogaard et al., 2021). In both works, it was found that PointNet++ was a suitable basis for semantic segmentation in the plant domain as compared to other deep neural networks.

4.2 Materials & Methods

In this section of the paper, we first present how the point-cloud data of the cucumber plants was obtained and labelled in section 4.2.1. In section 4.2.2, the concept of class imbalance is introduced. The method to create the training sets and how this method was used to battle class imbalance is explained in section 4.2.3. The training procedure is then presented in section 4.2.4 and the testing procedure and evaluation criteria are introduced in section 4.2.5.

4.2.1 Data acquisition

The cucumber plants used for this research were of the variety Proloog RZ F1 (Rijk Zwaan, De Lier, The Netherlands). Twelve plants were grown in a climate chamber on plant gutters. The plants were identified based on the location in the plant gutter as plant A1 up to B6, see Figure 4.2. An in-row distance between the plants of 1 m was used to prevent occlusion. The data acquisition period started when the plants had 8 leaves and a plant length of 76 cm on average. After 11 days, at the end of the data acquisition period, the average number of leaves was 12 and the average plant length was 195 cm. An impression of the plants is shown in Figure 4.1.

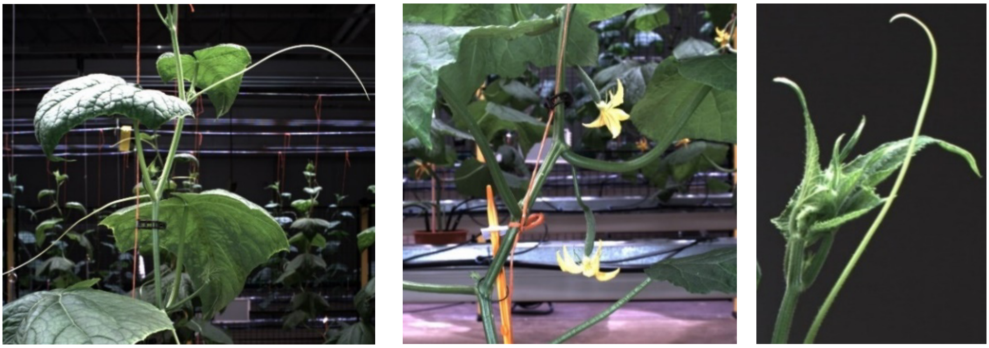


Figure 4.1 – Impression of the different parts of the cucumber plants. The left image contains leaves, stem, petioles, tendrils, and the supporting wire. In the middle image, two emerging fruits with flowers, in this research classified together as 'ovary', are visible. Also, some nodes are clearly visible in this image. The right image shows the growing point of the plant and another tendril.

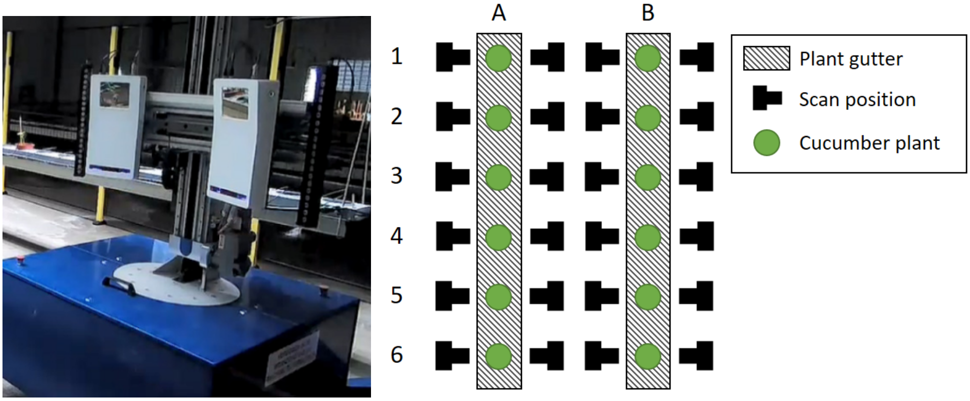


Figure 4.2 – The mobile platform (left) including the Phenospex F500 Dual scan system. The right image shows a schematic top-down overview of the plants in the climate chamber. The mobile platform automatically moved to the scan positions. At each position, the corresponding plant was scanned in a vertical direction by moving the scanners upwards. (Boogaard et al., 2021).

The point clouds used in this research were obtained using a Phenospex F500 Dual scan system (Phenospex, 2017), based on laser-line triangulation. A driving WIWAM plant analyser (WIWAM, 2022) (see Figure 4.2) was used to automatically position the scanners in front of each plant. The plants were then scanned in a vertical direction, from the plant gutter up to the growing point of the plant. Scans were made at both sides of the plant gutter. A schematic overview of the plants and the scanning positions is also given in Figure 4.2. The 24 scanning positions and the 11 days on which the plants were scanned resulted in a total dataset of 24 · 11 = 264 point clouds.

Next to the 3D spatial information, the Phenospex F500 Dual scan system also provided colour information. So, the input data contained six features for each point in the point clouds: *x*, *y*, and *z* for the 3D position of the point and red, green, and blue (*r*, *g*, and *b*) for the colour of the point.

To train PointNet++ and to be able to assess the quality of the trained network, a ground truth dataset was generated by manually segmenting all the 264 point clouds in the classes ‘stem’, ‘petiole’, ‘leaf’, ‘growing point’, ‘node’, ‘ovary’, ‘tendril’, and ‘non-plant’, using the segment module of CloudCompare (CloudCompare, 2019). The class ‘non-plant’ was used for the plant gutter, the pot, the plant label and the wire to which the plant was attached for support. An example of a coloured point cloud and a manually segmented point cloud is shown in Figure 4.3.

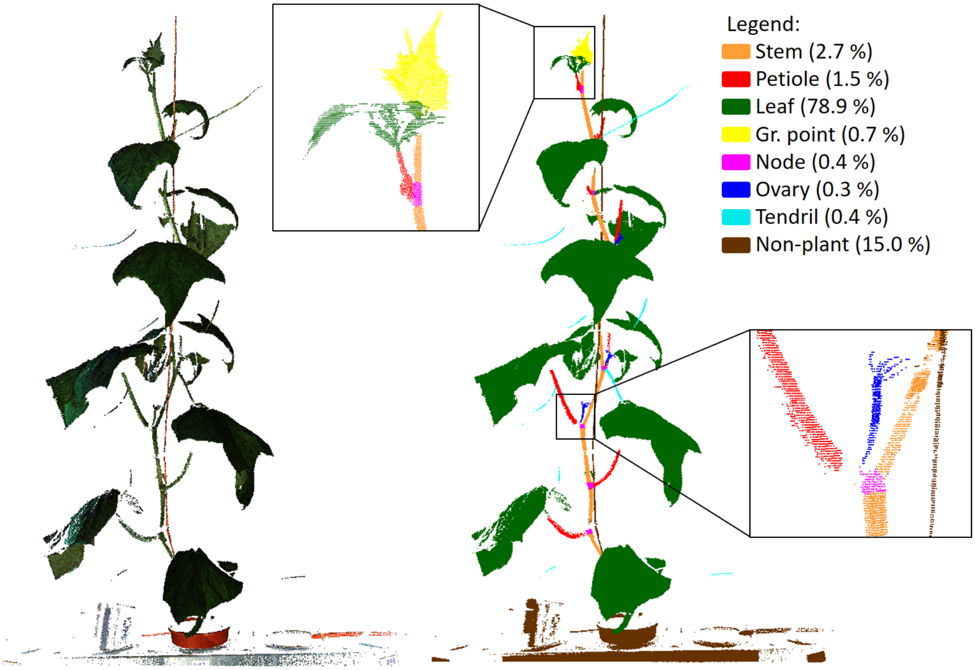


Figure 4.3 – Example of an original, coloured, point cloud (left) and a manually segmented point cloud, showing the 8 classes used in this research. In the legend, the fraction of points per class is given. The black squares show a zoomed in segment of the segmented point cloud.

4.2.2 Class imbalance

In Figure 4.3, the percentage of points in each of the classes is added in the legend of the figure, clearly indicating the imbalance in the dataset. Most of the points (78.9%) were 'leaf' points and also the 'non-plant' class was relatively large (15.0%). The classes 'stem' (2.7%) and 'petiole' (1.5%) were a lot smaller, while each of the remaining classes ('growing point', 'node', 'ovary', and 'tendril') contained less than 1% of the points.

The level of class imbalance was quantified using the Shannon entropy, according to Equation 4.1:

$$h(d) = -\sum_{c=1}^{C} p_c^d \cdot \log_2(p_c^d) \qquad \text{[bits]} \qquad \textit{Equation 4.1}$$

Where $h(d)$ is the entropy for dataset d, referring to either the manually labelled dataset, or one of the training sets created for the experiments done in this research. The number of classes is C and p_c^d is the probability that a randomly chosen point belongs to class c in dataset d, which is equal to the fraction of points in dataset d belonging to class c. The upper limit for the entropy can be found for a completely homogeneous dataset, which is the case when all classes are equally present in the dataset. For the 8 classes labelled in our dataset, the maximum value for the entropy was $-(8 \cdot \frac{1}{8} \cdot \log_2\left(\frac{1}{8}\right)) = 3$ bits. If a dataset is dominated by one of the classes, the entropy decreases and approaches 0 bits. The entropy of the manually segmented dataset, based on the class distribution as shown in Figure 4.3, was 1.1 bits.

4.2.3 Using the division into chunks to battle class imbalance

The process in which the original point clouds were divided into chunks was used as a mechanism to battle class imbalance in this work. In our procedure, the chunks were created in two steps. First, for each class, a pre-defined number of anchor points was sampled from the point cloud. An anchor point for a certain class was a randomly selected point from the set of all points of that class. In the second step, the neighbouring points were added to each of the anchor points, based on a *k*-nearest-neighbour search. This search was performed on the entire point cloud, including points of different classes than the class of the anchor point. The selected neighbourhood was then saved as a training sample. The value of *k* was first set at 4096 points, as this is a value often used in literature.

The first experiment to battle class imbalance focused on the selection of anchor points from the point cloud. The aim was to decrease the level of class imbalance by changing the number of anchor points per class, based on two strategies. For the first strategy, the anchor points were selected in a class-independent manner to maintain the original class-distribution as a reference. So, the number of anchor points per class was based on the original class distribution. For each class c, the number of anchor points a_c was defined by Equation 4.2, where n_c is the number of points in class c, n is the total number of points in the point cloud and a is the number of anchor points per point cloud. The value for a was set at 100, as a balance between processing time and the amount of data used for training.

$$a_c = \frac{n_c}{n} \cdot a \qquad \text{(class-independent)} \qquad [-] \qquad \textit{Equation 4.2}$$

The second training set was created using a class-dependent selection of anchor points. The hypothesis of this strategy was that a class-dependent selection of anchor points could reduce the level of class imbalance and improve the segmentation performance. For the class-dependent strategy, the number of anchor points per class was based on the inverse of the original distribution, according to Equation 4.3. First, the inverse fraction of points in a class was calculated as 1 minus the fraction of points in that class. The inverse fraction for each class was then divided by the sum of the inverse fractions for all classes, in order to obtain the fraction of anchor points for this class in the training set. The obtained fraction was multiplied by the total number of anchor points per point cloud, a, to obtain the number of anchor points per class. As in the previous strategy, the value for a was set at 100.

$$a_c = \frac{1 - \frac{n_c}{n}}{\sum_{i=1}^{C}(1 - \frac{n_i}{n})} \cdot a \qquad \text{(class-dependent)} \qquad [-] \qquad \textit{Equation 4.3}$$

The two resulting training sets are referred to as the class-independent and the class-dependent training set. The resulting class distribution in these two training sets is presented in Figure 4.4. For comparison, the class distribution of the manually labelled dataset is also shown. The entropy of the manually labelled data and the class-independent training set was both 1.1 bits, while the entropy of the class-dependent training set was 2.6 bits. This indicates that the class-dependent training set had a higher level of class balance.

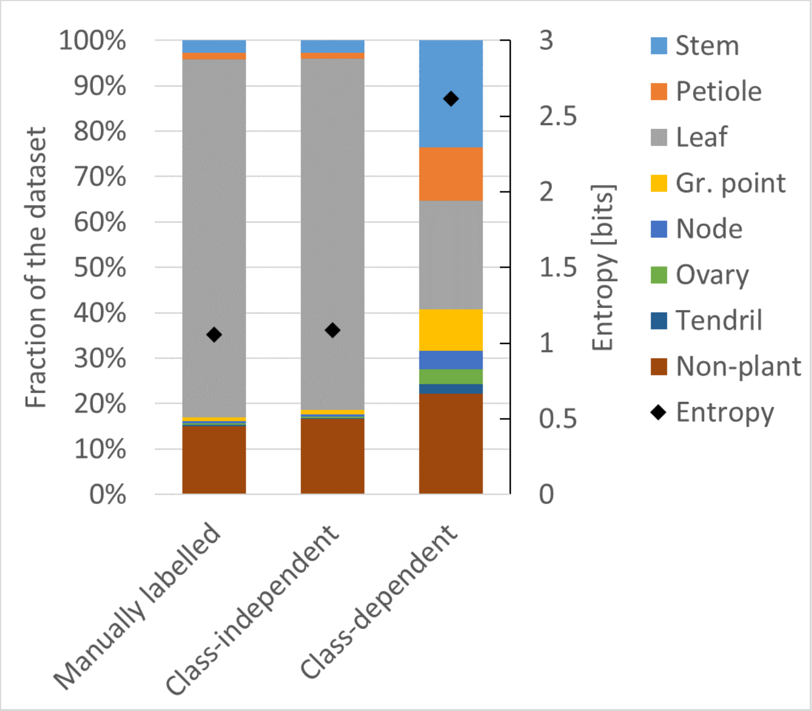


Figure 4.4 – Composition of the manually labelled dataset, the class-independent training set, and the class-dependent training set. The black diamonds show the level of entropy (secondary axis) according to Equation 4.1.

To provide more insight into how the two training sets differ from each other, the points that were selected for training for one point cloud are shown in Figure 4.5. The left image shows the selected points for the class-independent training set and the right image shows the selected points for the class-dependent training set. In the right image, the increased focus on the underrepresented classes (‘node’, ‘ovary’, ‘tendril’, but also ‘stem’ and ‘petiole’) and the decreased focus on the overrepresented classes (‘leaf’ and ‘non-plant’) is clearly visible.

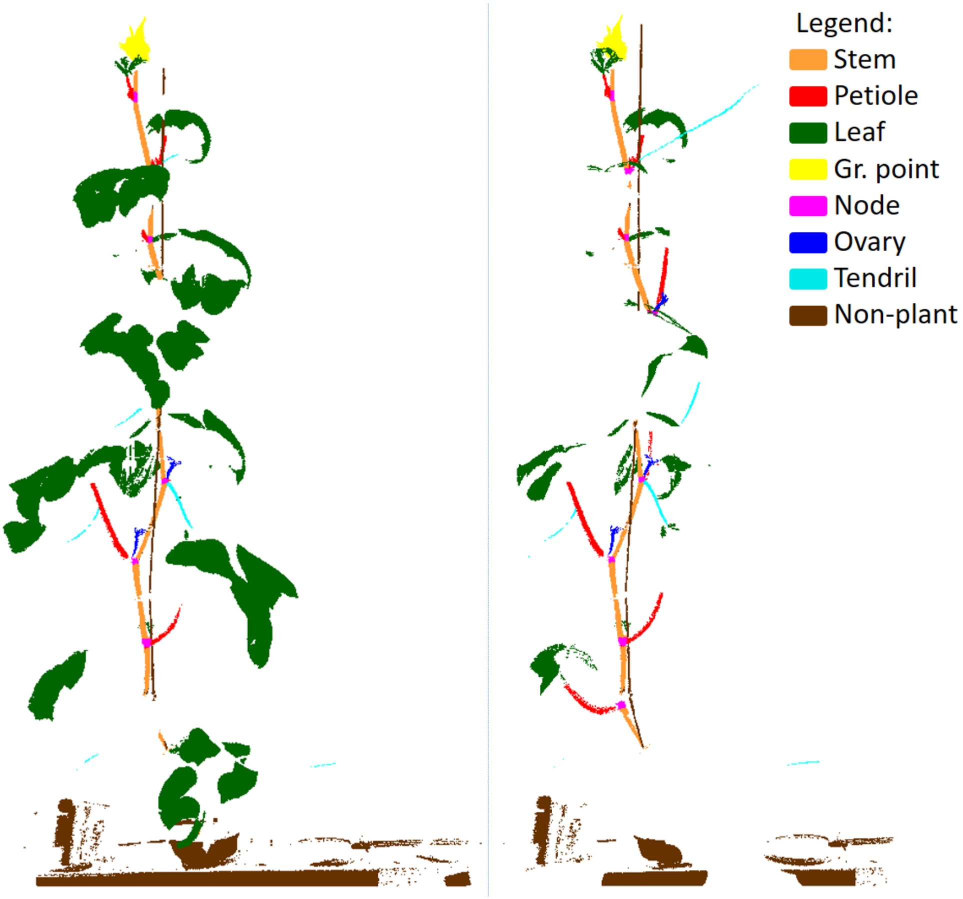


Figure 4.5 – Example of the selected points for the class-independent (left) and the class-dependent (right) training set. The class-dependent training set was more balanced. Note that points could be included multiple times in overlapping chunks, which is not visible in this figure.

Although the level of class balance in the class-dependent training set was increased, there was still quite some variation in the presence of classes. The smallest class (‘tendril’) occupied 2% of the points, while the largest class (‘leaf’) still occupied 24% of the points. The main reason was that the selected neighbourhood contained points of different classes than the class of the selected anchor point. The number of points in a chunk that were of a different class than the targeted anchor point depended on the physical size of the target class in the point cloud as well as the value of *k*. In fact, to obtain a completely homogeneous dataset, an equal amount of anchor points from each class could be selected and the chunk size could be set to 1, to prevent the inclusion of neighbouring points of different classes. However, this would also cause the complete loss of the context of the selected anchor point. Neighbouring points add information about local point densities and geometrical aspects like the curvature of the plant part. To provide an intuition about why this is important, consider that an individual point contains only six features: its x, y, and z location and the r, g, and b colour. If only these six features would be known, especially in the current data, it is infeasible to know to what class a point belongs. The features of neighbouring points help to predict the correct class.

This apparent tension between class balance and context was further investigated in a second experiment. We created additional training sets based on the class-dependent strategy, using *k = 512, k = 1024, k = 2048, k = 4096, k = 8192,* and *k = 16384*. The number of anchor points per point cloud, a, remained 100 for each of these training sets. The training set for *k = 4096* was the same as in experiment 1. The composition and the entropy for the generated training sets are shown in Figure 4.6. Indeed, the entropy for the smaller chunk sizes was higher than the entropy for the larger chunk sizes, indicating that the training sets for smaller values of *k* have a lower level of class imbalance. The variation in presence between the classes was lowest at *k = 512*, ranging from 7% (‘ovary’ and ‘tendril’) to 23% (‘stem’).

Besides context in the sense of the number of neighbourhood points (*k*), another way to look at the context of a point is to count the number of classes that were present in a chunk. The number of classes per chunk was analysed for the different values of *k*, as shown in Figure 4.7. The mean value for the number of classes per chunk increased for larger chunk sizes, indicating that a higher level of context was present in larger chunks.

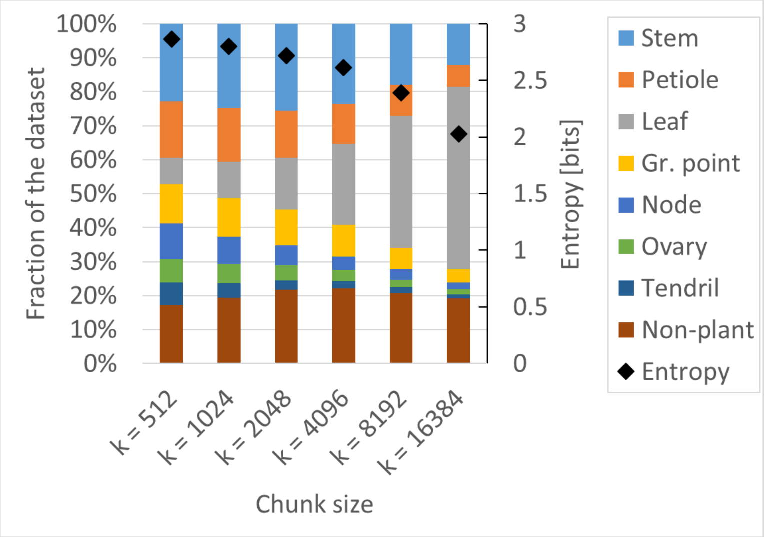


Figure 4.6 – Composition of the six training sets created for experiment 2. All datasets are based on the class-dependent sampling strategy using six different values for k. *The black diamonds shows the level of entropy (secondary axis) according to Equation 4.1.*

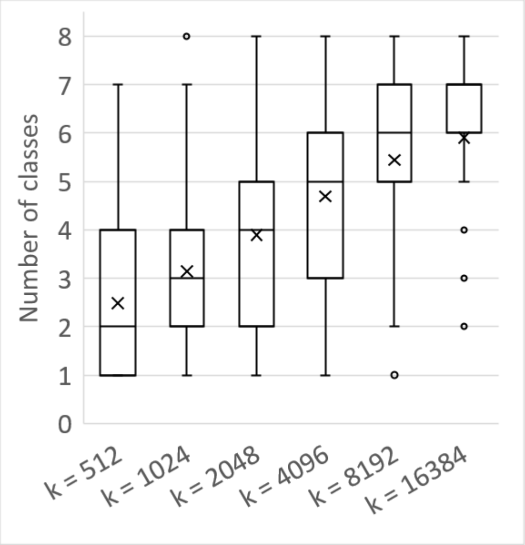


Figure 4.7 – Boxplot showing the number of classes per chunk for different values of k, based on the training sets created using $a = 100$. The mean values are shown as 'x'. The box shows the lower and upper quartile, the whiskers indicate the highest and lowest number of classes per chunk within 1.5 times the inter-quartile range. Values outside this range were considered outliers and are shown as a circle.

In summary, a class-independent training set was generated for *k = 4096* and six class-dependent training sets were generated for *k = 512, k = 1024, k = 2048, k = 4096, k = 8192,* and *k = 16384*. The training sets were used to train PointNet++, as presented in the next section.

4.2.4 Training procedure

The PointNet++ implementation published by Qi et al. (2018) was trained to segment the chunks of the point clouds into plant parts. The manually labelled data was first split into a training, validation, and test set. To prevent that data from one plant was in more than one of these sets, this split was made on a plant level. As the result of a random selection procedure, all data of plant B6 were selected as the validation set and all data of plant A1 were selected as the test set (see Figure 4.2 for the plant IDs). In previous work using the same dataset (Boogaard et al., 2021), a cross-validation was done to quantify the variation in performance when different plants would be used for the training, validation, and test split. For the population of plants considered, it was found that variation was low. Therefore, in the present research, no cross-validation was performed.

The aim of the network was to predict to which plant part each point belongs, generally called semantic segmentation. Besides features on a per point basis, the network calculated contextual features depending on the provided neighbourhood points. To obtain these features, a set-abstraction level was used, which consisted of three layers. In the sampling layer, a set of points was selected from the input chunk using a farthest point sampling algorithm. These points were grouped with points in their neighbourhood in the grouping layer. Finally, for each of the groups, a PointNet-based (Qi et al., 2016) layer was applied to learn the local point features. The set abstraction was then repeated to learn features on a larger scale. Because of the sampling and grouping, not all points were processed in this way. The feature vectors for unprocessed points were obtained by distance-based interpolation. No changes were made to the PointNet++-architecture in our study, so for further details about the network we refer to the work of Qi et al. (2016, 2018).

The training procedure used a sparse softmax cross-entropy loss. As mentioned in the introduction, the loss function could be changed to increase the loss for incorrect predictions of minority classes. Since we focused on the effect of improved class balance in the training set, we did not change the loss function in the current research. After each training epoch, the loss on the validation set was monitored and training was stopped when this loss no longer decreased. The weights that corresponded to the lowest observed loss on the validation set were then used for further evaluation of the performance.

Since the training of a neural network is a stochastic process, the results can slightly differ per training run and therefore, the training was repeated five times for each configuration. The results, presented in section 4.3, are based on the average performance of these repetitions.

4.2.5 Testing procedure and evaluation

As mentioned in section 4.2.4, all point clouds of plant A1 were used as the test set. This means that the data of this plant was not used in the training nor in the validation of the method, such that the performance of the method on this plant resembled the performance on new data. Similar to the training sets, the point clouds of the test set were split into chunks. Since the objective of the test set was to estimate the performance on new data, meaning that no manually assigned labels would be available, the class-independent sampling approach was used for the test set. As a consequence, the class distribution in the test set matched the class distribution in the original data.

Furthermore, to be able to segment the entire point clouds of the test set, it was necessary to make sure that all points were included in at least one of the chunks. Therefore, the number of chunks per point cloud, a, was not set on forehand for the test set. Instead, the selection of anchor points was repeated until all points of the point cloud had been added to at least one chunk. In this procedure, the anchor points were iteratively selected from the set of points that were not yet added to a chunk. To maintain local structures and point densities, the k-nearest-neighbours were selected from the set of all points.

The trained network was then used to predict the segmentation of all chunks in the test set. The segmented chunks were merged to obtain the segmentation of the entire plant. Due to the way the chunks were created, points could be present in multiple chunks. If multiple predictions were present for a point, the predicted class in the merged point cloud was based on a majority-voting strategy including all individual predictions for that point. In case the voting ended in a tie, one of the classes in this tie was selected at random. The merged point clouds were used for the quantitative evaluation of the experiments. That is, the evaluation was done on the full point clouds of the plants.

The segmentation performance obtained in the experiments is reported as the intersection over union (IoU) between the manually labelled data and the predictions of the trained network for the test set. The IoU was based on a point-wise comparison between the manual and predicted point labels. A point was considered a true positive (TP) if the predicted label was equal to the manual label.

If the predicted label was not equal to the manual label, a point was considered a false positive (FP) for the predicted class and a false negative (FN) for the manually assigned class. Based on the number of TP, FP, and FN points, the IoU for class c was calculated according to Equation 4.4:

$$\mathrm{IoU_c} = \frac{\mathrm{TP_c}}{\mathrm{TP_c} + \mathrm{FP_c} + \mathrm{FN\,_c}} \quad [\text{-}] \qquad \textit{Equation 4.4}$$

Besides the IoU-values per class, the average IoU-value is reported in two ways. First, the micro average of the IoU was calculated according to Equation 4.5. This provided an indication of the IoU for the entire dataset on a point level. However, since the 'leaf' was heavily overrepresented in the test set, this value was dominated by the IoU of the class 'leaf'. Therefore, we also report the macro average, which is the average of the per-class IoU-values. The macro average was calculated according to Equation 4.6.

$$\mathrm{IoU_{micro}} \; = \frac{\sum_{\mathrm{c}=1}^{\mathrm{C}} \mathrm{TP}_c}{\sum_{\mathrm{c}=1}^{\mathrm{C}}(\mathrm{TP}_c + \mathrm{FP}_c + \mathrm{FN}_c)} \quad [\text{-}] \qquad \textit{Equation 4.5}$$

$$\mathrm{IoU_{macro}} \; = \frac{\sum_{\mathrm{c}=1}^{C} \mathrm{IoU}_c}{C} \quad [\text{-}] \qquad \textit{Equation 4.6}$$

Where C is the total number of classes.

The precision (P) and recall (R) are also reported per class, based on Equation 4.7 and Equation 4.8. The precision reports what proportion of points that was predicted as a certain class, actually belonged to that class. The recall reports what proportion of points that actually belonged to a certain class, was also predicted as that class.

$$\mathrm{P_c} = \frac{\mathrm{TP_c}}{\mathrm{TP_c} + \mathrm{FP_c}} \quad [\text{-}] \qquad \textit{Equation 4.7}$$

$$\mathrm{R_c} = \frac{\mathrm{TP_c}}{\mathrm{TP_c} + \mathrm{FN_c}} \quad [\text{-}] \qquad \textit{Equation 4.8}$$

The significance of differences between reported IoU-values was tested using a Wilcoxon signed-rank test. The outcome of these tests is reported as 'n.s.' when the difference was not significant, as * when p<0.05, as ** when p<0.01, and finally as *** when p<0.001.

4.3 Results

In this section, the results of the two experiments are presented. First, in section 4.3.1, the segmentation performance for the class-independent and the class-dependent strategy used to select the anchor points is reported. In section 4.3.2, the performance for the different chunk sizes is presented, based on the class-dependent sampling strategy. All results are based on the test set.

4.3.1 Selection of anchor points

The performance of the segmentation for the two sampling strategies, both with chunk size 4096, is shown in Figure 4.8. When comparing the class-dependent sampling strategy to the original sampling strategy, for most classes and the average IoU-values, a significant improvement of the segmentation quality was observed. Exceptions are the ‘growing point’ (no significant difference) and the ‘leaf’, where a very small but significant decrease of the IoU was observed.

The biggest improvement was observed for the smallest classes, as they suffered most from the imbalance in the original dataset. The IoU increased from 0.02 to 0.34 for the ‘node’, from 0.24 to 0.48 for the ‘ovary’ and from 0.30 to 0.48 for the ‘tendril’. The IoU values for the ‘stem’ and ‘petiole’ also increased, from 0.65 to 0.79 and from 0.58 to 0.76 respectively. Finally, the micro average of the IoU increased from 0.95 to 0.96 and the macro average of the IoU increased from 0.54 to 0.68, for the class-dependent sampling strategy.

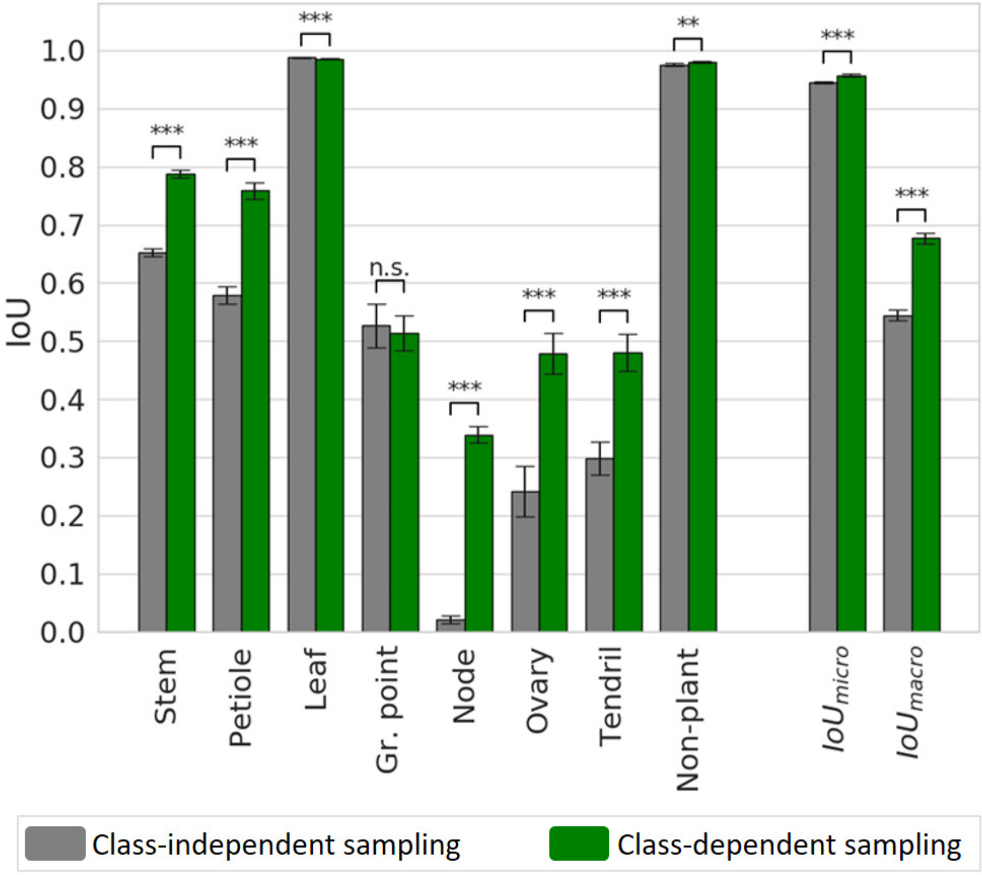


Figure 4.8 – IoU for the class-independent and the class-dependent sampling strategy based on the test set. Bars indicate the mean IoU and the error bars give the 95% confidence interval on the mean. The arches indicate the significance of the differences between the observed IoU-values.

The confusion matrices for the class-independent and class-dependent sampling strategy as well as the difference between the two are reported in Table 4.1. For the class-independent training set, the majority classes showed the highest percentage of correctly predicted points. For the classes ‘leaf’ and ‘non-plant’, the percentage of correctly predicted points, shown on the diagonal, was 99.5% and 98.6% respectively. Also the ‘stem’, ‘petiole’, and ‘growing point’ showed a high percentage of correctly predicted points, 86.7%, 73.4%, and 92.5% respectively. The performance was lower for ‘ovary’ (50.0%) and ‘tendril’ (32.9%). For the class ‘node’, only 2.7% of the points was correctly classified by the network.

The values outside the diagonal show what percentage of points that was manually labelled as a certain class, was predicted as a different class. The most errors were made for ‘node’ points, 69.0% of these points was predicted as ‘stem’ and 21.3% was predicted as ‘petiole’. Also ‘ovary’ and ‘tendril’ were often incorrectly predicted as ‘stem’ or ‘petiole’. Furthermore, for the class ‘tendril’, 21.4% of the points was predicted as ‘leaf’.

When training on the class-dependent training set, the percentage of correctly predicted points increased for most of the classes. The percentage of ‘node’ points that was correctly classified by the network increased by 46.0 percentage points to 48.7%. This improvement was mainly due to fewer ‘node’ points being predicted as ‘stem’ and ‘petiole’. On the other hand, the number of ‘stem’, ‘petiole’, ‘ovary’, and ‘tendril’ points incorrectly predicted as ‘node’ also increased slightly. The confusion between ‘stem’, ‘petiole’, and ‘leaf’ was strongly reduced. The percentage of ‘node’, ‘ovary’ and ‘tendril’ points that were predicted to be ‘stem’ and ‘petiole’ was also reduced, although still 38.9% of the ‘node’ points was classified as ‘stem’ and 23.2% of the ‘ovary’ points was classified as ‘petiole’. Only for the class ‘leaf’, a decrease in the number of correctly classified points was observed of 0.5 percentage point. Still, 99.0% of the ‘leaf’ points was correctly classified and the number of false positives for the ‘leaf’ decreased.

The precision and recall for the network trained on the class-independent and class-dependent training set are reported in Table 4.2. For most classes, the precision and the recall were higher for the class-dependent training set. The recall for ‘leaf’ and the precision for ‘growing point’ were slightly lower for the class-dependent training set. For the class ‘tendril’, the precision for the class-dependent training set is 16 percentage points lower than for the class-independent training set. This was mostly due to ‘leaf’ points incorrectly predicted as ‘tendril’.

Table 4.1 – Confusion matrices for the trained network using the class-independent selection of anchor points, the class-dependent selection of anchor points and the difference between the two. Each row in the confusion matrices is based on the points that were labelled as that class by hand. The percentages indicate how these points were predicted by the network, meaning that values on the diagonal are correct predictions (marked in bold, also known as recall). Wrong predictions are highlighted in red and correct predictions are highlighted in green, brighter colours correspond to higher values.

Class-independent **(percentage)**		Predictions							
		Stem	Petiole	Leaf	Gr. point	Node	Ovary	Tendril	Non-plant
True labels	Stem	**86.7**	6.5	1.9	1.5	0.4	0.7	0.1	2.1
	Petiole	17.2	**73.4**	6.4	1.3	0.3	0.9	0.3	0.2
	Leaf	0.0	0.0	**99.5**	0.4	0.0	0.0	0.0	0.0
	Gr. Point	0.3	0.0	5.9	**92.5**	0.0	0.0	0.2	1.0
	Node	69.0	21.3	1.6	2.2	**2.7**	1.4	0.1	1.8
	Ovary	20.9	24.9	1.8	0.2	0.9	**50.0**	0.8	0.5
	Tendril	21.8	7.9	21.4	1.1	0.2	2.7	**32.9**	12.0
	Non-plant	0.7	0.0	0.5	0.1	0.0	0.0	0.0	**98.6**

Class-dependent **(percentage)**		Predictions							
		Stem	Petiole	Leaf	Gr. point	Node	Ovary	Tendril	Non-plant
True labels	Stem	**91.1**	1.7	0.5	0.9	3.2	0.5	0.2	1.9
	Petiole	3.2	**88.7**	3.2	0.7	2.9	0.6	0.6	0.1
	Leaf	0.1	0.2	**99.0**	0.5	0.0	0.0	0.1	0.1
	Gr. point	0.9	0.1	1.9	**96.1**	0.1	0.0	0.2	0.7
	Node	38.9	7.8	0.7	1.2	**48.7**	0.7	0.2	1.9
	Ovary	4.3	23.2	1.0	0.1	5.0	**65.4**	1.0	0.1
	Tendril	10.3	3.9	8.4	0.8	2.2	0.5	**63.8**	10.2
	Non-plant	0.5	0.0	0.2	0.1	0.0	0.0	0.1	**99.0**

Table 4.1 – Continued

Difference **(percentage point)**		Predictions							
		Stem	Petiole	Leaf	Gr. point	Node	Ovary	Tendril	Non-plant
True labels	Stem	**4.4**	-4.8	-1.5	-0.6	2.8	-0.2	0.1	-0.2
	Petiole	-14.0	**15.3**	-3.2	-0.6	2.6	-0.3	0.3	-0.1
	Leaf	0.1	0.1	**-0.5**	0.2	0.0	0.0	0.1	0.0
	Gr. point	0.6	0.1	-4.0	**3.6**	0.1	0.0	0.0	-0.3
	Node	-30.1	-13.5	-0.9	-1.0	**46.0**	-0.7	0.1	0.1
	Ovary	-16.6	-1.7	-0.8	-0.1	4.0	**15.4**	0.2	-0.3
	Tendril	-11.6	-4.0	-13.1	-0.3	2.0	-2.2	**30.8**	-1.8
	Non-plant	-0.2	0.0	-0.2	0.0	0.0	0.0	0.1	**0.4**

Table 4.2 – Precision (P) and Recall (R) per class, for the class-independent and the class-dependent selection of anchor points. Per column, the highest value is highlighted in green and the lowest value is highlighted in red.

	Stem		Petiole		Leaf		Gr. point		Node		Ovary		Tendril		Non-plant	
	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R
Class-independent	0.72	0.87	0.74	0.73	1.00	1.00	0.55	0.93	0.34	0.03	0.80	0.50	0.85	0.33	0.99	0.99
Class-dependent	0.85	0.91	0.83	0.89	1.00	0.99	0.51	0.96	0.55	0.49	0.89	0.65	0.69	0.64	0.99	0.99

4.3.2 Number of points per chunk

The results of the experiment to test the effect of different chunk sizes on the segmentation performance are presented in this section. In Figure 4.9, the IoU-values including the 95% confidence intervals on the mean are plotted. The significance of the differences for all classes are reported in Table 4.A1 in appendix A, based on a two-sided test.

For most classes, there seems to be an optimal chunk size at one of the intermediate values. For ‘stem’, the highest IoU was observed at *k = 2048*, for ‘petiole’ the highest IoU was observed for *k = 8192*. For these classes the variation in IoU between *k = 2048*, *k = 4096,* and *k = 8192* was very small. For the classes ‘node’ and ‘ovary’, the IoU was higher for smaller chunk sizes with a maximum at *k = 1024*. However, for *k = 512*, the IoU was lower at a similar level as for *k = 4096*. A similar pattern was found for the ‘tendril’, having the highest IoU at *k = 4096*. The highest IoU for the class ‘non-plant’ was found at *k = 8192*, although this value was only significantly higher than the IoU for *k = 16384*. For the classes ‘leaf’ and ‘growing point’, the highest value was found at *k = 16384*. However, it could be the case that, if even larger values for *k* were added to the experiment, also for these classes the IoU would drop. On average, the results did show an optimal value at *k = 8192* for the micro average and at *k = 4096* for the macro average of the IoU.

To provide more insight into the precision-recall trade-off, the precision and recall for each class are reported in Table 4.3. For ‘stem’ and ‘petiole’, both the precision and the recall decreased for smaller chunk sizes, caused by a higher fraction of FP and FN for these classes. The precision for the ‘leaf’ stayed very high, meaning that if a point was classified as ‘leaf’, it almost certainly was correct. On the other hand, the recall for the ‘leaf’ decreased for smaller chunk sizes, meaning that no longer all ‘leaf’ points were retrieved for these chunk sizes. Based on visual inspection of the segmented point clouds, the main reason was identified as ‘leaf’ points incorrectly classified as ‘growing point’. This also caused the drop in precision for the class ‘growing point’.

Looking at the classes ‘node’ and ‘ovary’, the changes in precision and recall showed an optimum chunk size around 2048 or 1024 points. The optimum chunk size for ‘tendril’ was 16384 when aiming for precision, while it was 512 when aiming for recall. Finally, the precision and recall for the class ‘non-plant’ were high and stable among the different chunk sizes.

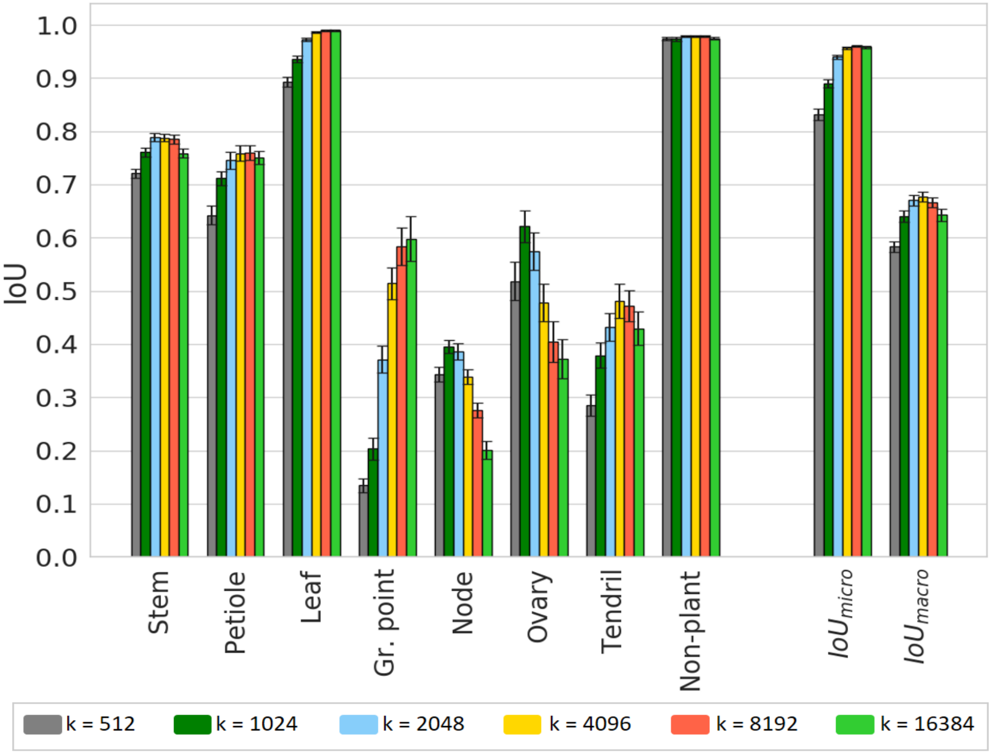


Figure 4.9 – IoU for the test set using six different chunk sizes, based on the class-dependent sampling strategy. Bars indicate the mean IoU and the error bars give the 95% confidence interval on the mean. The significance of differences between the observed IoU-values are reported separately in Table 4.A1 in Appendix A for readability.

Table 4.3 – Precision (P) and Recall (R) per class, for the six different chunk sizes. Values highlighted in green correspond to the higher values per column, while values highlighted in red correspond to the lower values per column.

	Stem		Petiole		Leaf		Gr. point		Node		Ovary		Tendril		Non-plant	
Chunk size:	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R
512	0.82	0.85	0.74	0.82	1.00	0.90	0.07	0.96	0.49	0.53	0.74	0.80	0.32	0.77	0.99	0.99
1024	0.84	0.89	0.80	0.85	1.00	0.94	0.12	0.96	0.56	0.58	0.90	0.79	0.46	0.76	0.98	0.99
2048	0.86	0.91	0.83	0.87	1.00	0.98	0.29	0.97	0.58	0.57	0.89	0.75	0.54	0.71	0.99	0.99
4096	0.85	0.91	0.83	0.89	1.00	0.99	0.51	0.96	0.55	0.49	0.89	0.65	0.69	0.64	0.99	0.99
8192	0.85	0.91	0.84	0.88	1.00	0.99	0.61	0.94	0.50	0.39	0.87	0.60	0.77	0.59	0.99	0.99
16384	0.82	0.92	0.84	0.87	1.00	1.00	0.64	0.89	0.46	0.29	0.88	0.55	0.78	0.54	0.99	0.99

4.4 Discussion & recommendations

4.4.1 Data set

The cucumber plants used in this research were grown in a climate chamber at an in-row distance of 1 meter. The distance between the plants prevented occlusion of plants by neighbouring plants. The main advantage of this experimental design was that each point cloud contained data of only one plant, which reduced the complexity of the segmentation task. For plant-phenotyping experiments, especially in data-acquisition systems based on a plant-to-sensor concept, it is a reasonable assumption that individual plants can be scanned. However, besides plant scientists and breeders, the presented methodology might be of interest for commercial growers, to assess the current state of their crop. When measuring plants in a denser and more complex environment, resulting in plant-plant occlusions, for each plant part it needs to be identified to which plant it belongs. This is an additional task, for which suitable methods need to be developed. The current dataset does not provide sufficient variation in complexity of the plants and their environment to study the effect of the level of complexity in the data on the segmentation performance. Extending the dataset in this direction would be valuable to further investigate practical applicability of the method for isolated plants as well as for high-density growing systems encountered in horticultural practice.

Although the measurement set-up resulted in high-resolution point clouds of individual plants, the point clouds were still not complete. That is, parts of the plant were missing in the point cloud, due to occlusion by other plant parts from the viewpoint of the scanner. Since PointNet++ relies on local geometrical features to segment the point clouds, it is likely that the missing data had a negative effect on the segmentation performance. A more in-depth investigation of this effect is recommended as future research. For this research, the level of completeness was the same for all experiments and it is not likely that it effected our results on dealing with class imbalance. We expect that the conclusions drawn at the end of the paper would not change if a more complete dataset was used.

The results presented in this paper clearly show that the class-dependent strategy outperformed the class-independent strategy. In this comparison, the class-independent strategy, with a chunk size of 4096 points, was used as a baseline. Although it was shown in Figure 4.4 that the class distribution of the class-independent training set indeed matched the original class distribution, the training set was constructed using our sampling procedure instead of a sliding window approach. This means that also for the class-independent training set, not all points of the original point clouds were used for training, contrary to current practice. To further assess the validity of this training set as a baseline, the performance of the method trained on the class-independent training set was compared to the performance reported by Boogaard et al. (2021). In that work, a common sliding window approach was used and all points of the original point clouds were included in the chunks. The average difference between the IoU-values of the classes was only 1 percentage point, and the micro and macro averages of the IoU were equal, indicating that the current class-independent training set provided a suitable baseline.

4.4.2 Sampling of the training data

Seven training sets were created for the experiments, the class-independent training set for *k = 4096* and six class-dependent training sets for *k = 512, k = 1024, k = 2048, k = 4096, k = 8192,* and *k = 16384.* The resulting training sets depended on three main parameters: the number of anchor points per class (a_c), the number of anchor points per point cloud (a) and the method to define the neighbourhood of the anchor point, which was selected as the chunk. The effects of these parameters are discussed in the following three paragraphs.

First, in the current work, the number of anchor points per class was based on the inverse of the class distribution, instead of using an equal amount of anchor points per class. This resulted in a set of anchor points that contained many points from the minority classes such as ‘node’ and ‘ovary’, and only few points from the majority classes, such as ‘leaf’. With a hypothetical setting of *k = 1*, this would result in a training set that would also be imbalanced but reversed to the original imbalance. This effect can be seen for *k = 512* in Figure 4.6, where the original majority class ‘leaf’ is now a minority class. Still, the training set based on *k = 512* had the highest level of entropy, meaning that it was the most balanced training set. This is caused by the inclusion of neighboring points, which were often from the original majority classes.

Second, the total number of anchor points per point cloud was set to 100 in the current work. The reason to keep this value constant was to obtain an equal amount of training samples, or chunks, for each of the values of *k*. However, instead of considering each chunk as a training sample, each individual point could be seen as a training sample. From that perspective, to keep the number of training samples constant, not the number of chunks, but the number of points in the training set ($a \cdot k$) should be kept constant. In our case, the number of points in the training set increased for larger values of *k*. Although in general more training data leads to a better performance, the best performance for all classes except ‘leaf’ was found at lower values of *k*, indicating the importance of class balance in the training set. It is recommended to further investigate the optimal number of anchor points per point cloud in relation to the chunk size. Besides saving on computation time, using less chunks also means that the effort needed for manually labelling the data can be reduced.

The third parameter was the algorithm used to select the neighbourhood of the anchor points. In this work, a nearest neighbour search was used, meaning that all selected points fitted within a sphere. In other applications, for example focusing on elongated objects, differently shaped neighbourhoods might be more suitable. Although different choices could have been made for the three parameters that were discussed, the conclusions in the light of the current research remain valid. It was clearly shown that the proposed class-dependent sampling method to create the training set did improve the segmentation quality.

4.4.3 Evaluation

To evaluate the proposed method, the trained networks were used to segment the point clouds of a test set. To construct this test set, anchor points were added until all points of the point cloud were part of at least one chunk. This method can be relatively inefficient, as points can be included in multiple chunks. Based on the results of the majority-voting strategy to reconstruct the plants after segmentation, it turned out that having multiple predictions per point did not drastically improve the results, although this was not thoroughly investigated. Including fewer points in multiple chunks can reduce processing time. Therefore, an alternative approach to create the test set could be to apply a clustering algorithm that divides the input point cloud into clusters of the chosen chunk size. Unfortunately, most clustering algorithms do not guarantee a fixed number of members per cluster. A possible solution was presented by Yi and Moon (2014), who did propose a k-means based clustering method with a fixed number of cluster members. This and other alternatives to optimise the test set should be further investigated.

The evaluation of the segmented test set was based on the IoU, the precision, and the recall. These are common evaluation criteria for semantic-segmentation methods and the results show that our proposed method outperformed the baseline with respect to these criteria. However, although the segmentation of the minority classes did improve, the performance for these classes is still lagging behind the performance for the majority classes. There is no general definition of when a segmentation method is good enough and therefore it is difficult to assess the value of the proposed improvement. We recommend to evaluate the current state-of-the-art plant-segmentation methods not only in the light of computer-vision based criteria, like IoU, precision, and recall, but also in the light of plant-based criteria like internode length, leaf angle, and leaf area. This requires that during the data-acquisition phase of the phenotyping experiment, sufficient manual measurements of such plant traits need to be added to the dataset. Unfortunately, for the current data set, these measurements were not available.

4.4.4 Final recommendations

In this research, we focused on the improvement of the segmentation quality that could be achieved using dedicated sampling of the training data. However, as mentioned in the introduction, the training procedure itself also provides opportunities to improve the segmentation performance on minority classes, for instance using data augmentation, weighted loss functions, or synthetic data. Also increasing the amount of training data, possibly using multiple annotations per input sample, as suggested in (Boogaard et al., 2021), could improve the segmentation. Evaluating these aspects goes beyond the scope of the current paper. We recommend to further explore these opportunities, quantifying the individual as well as the combined contributions to an improved segmentation method, in future work.

Finally, the class-dependent sampling method for creating the chunks was set-up in a generic way and is in principle not limited to cucumber plants. Specific settings, like the values for k and a, might be application or dataset specific. Therefore, the effect of these settings needs to be investigated to select a suitable value, similar to the way other hyper parameters of the network are set. The main procedure can then be tested on any point-cloud segmentation task. As imbalanced data is a typical problem in point cloud processing, also outside the plant domain, it would be interesting to test if the proposed method to create more balanced training sets also improves the segmentation performance in other applications and datasets.

4.5 Conclusion

We have presented a method to battle class imbalance in the segmentation of point clouds of cucumber plants. As hypothesized, the results of the first experiment showed that the segmentation performance on the original, imbalanced, test data was significantly improved using a more balanced training set. As expected, the biggest improvements were found for the smallest classes. The percentage of correctly predicted ‘node’ points increased by 46.0 percentage points, followed by the ‘tendril’, for which the percentage of correctly predicted points increased by 30.8 percentage points. Also the overall segmentation quality improved, with the micro average of the IoU increasing from 0.94 to 0.96 and the macro average of the IoU increasing from 0.54 to 0.68.

In the second experiment, the trade-off between the amount of context and the class balance was investigated. Six training sets were created for different values of the chunk size, *k.* This parameter defined the number of points sampled around each anchor point. Lower values for *k* tended to increase the level of class balance in the training set, while higher values added more context around the anchor point. The results showed that higher levels of class balance did not necessarily lead to better segmentation performance. The value for *k* showing the best segmentation results differed per class. Based on the average IoU-values, the best segmentation results were obtained using *k = 8192* for the micro average or using *k = 4096* for the macro average.

In this paper, we have demonstrated that class-dependent sampling of the training data to improve the class balance in the training set led to an improved point-cloud segmentation method for plant phenotyping.

4.6 Appendix A

Table 4.A1 – Significance table, showing if the difference between IoU-values observed for different chunk sizes was significant, using a two-sided test. Which of the chunk sizes had a higher IoU can be seen in Figure 4.9. The table is symmetrical, for readability only the values below the diagonal are printed.

	Chunk size	512	1024	2048	4096	8192
Stem	1024	***				
	2048	***	***			
	4096	***	***	n.s.		
	8192	***	***	n.s.	n.s.	
	16384	***	n.s.	***	***	***
Petiole	1024	***				
	2048	***	***			
	4096	***	***	n.s.		
	8192	***	***	n.s.	n.s.	
	16384	***	***	n.s.	n.s.	n.s.
Leaf	1024	***				
	2048	***	***			
	4096	***	***	***		
	8192	***	***	***	***	
	16384	***	***	***	***	n.s.
Gr. point	1024	***				
	2048	***	***			
	4096	***	***	***		
	8192	***	***	***	***	
	16384	***	***	***	***	n.s.
Node	1024	***				
	2048	***	n.s.			
	4096	n.s.	***	***		
	8192	***	***	***	***	
	16384	***	***	***	***	***
Ovary	1024	***				
	2048	***	**			
	4096	**	***	***		
	8192	***	***	***	***	
	16384	***	***	***	***	*

Table 4.A1 – Continued

	Chunk size	512	1024	2048	4096	8192
Tendril	1024	***				
	2048	***	***			
	4096	***	***	***		
	8192	***	***	***	n.s.	
	16384	***	***	n.s.	***	***
Non-plant	1024	n.s.				
	2048	***	**			
	4096	*	n.s.	n.s.		
	8192	n.s.	n.s.	n.s.	n.s.	
	16384	n.s.	n.s.	***	***	***
$\text{IoU}_{\text{micro}}$	1024	***				
	2048	***	***			
	4096	***	***	***		
	8192	***	***	***	***	
	16384	***	***	***	n.s.	***
$\text{IoU}_{\text{macro}}$	1024	***				
	2048	***	***			
	4096	***	***	n.s.		
	8192	***	***	n.s.	**	
	16384	***	n.s.	***	***	***

Chapter 5

The added value of 3D point clouds for digital plant phenotyping

a case study on internode length measurements in cucumber

This chapter has been submitted and is under review as:

Boogaard, F. P., van Henten, E. J., & Kootstra, G., The added value of 3D point clouds for digital plant phenotyping – a case study on internode length measurements in cucumber, *Biosystems Engineering (under review)*

Nomenclature

Point features	r, g, b	Spectral features: red, green, and blue
	x, y, z	Geometric features (x, y, z location of point)
Evaluation	s_i	Ground-truth internode length between node *i* and node *i+1* [mm]
	$\hat{\mathrm{s}}_i$	Estimated internode length between node *i* and node *i+1* [mm]
	e_i	Error between estimated internode length and ground-truth internode length [mm]
	RMSE	Root Mean Square Error [mm]
	$\mathrm{IoU_c}$	Intersection-over-Union for class c
	$\mathrm{TP}, \mathrm{FP}, \mathrm{FN}$	True positive, False positive, False negative
	P, R	Precision, Recall
	F1	F1-score, the harmonic mean of precision and recall

Abstract

Computer-vision based methods contribute to the availability of high-quality phenotypic datasets. Most computer-vision based methods for plant phenotyping are based on analysis of 2D images. However, previous research showed that for traits related to plant architecture, like internode length, a main limitation of 2D methods was that plants with a curved growing pattern could not be accurately measured. In this work, it was hypothesized that methods based on 3D data can overcome this limitation, while increasing the overall accuracy of the internode length measurements.

To test the hypothesis, we propose a method to estimate internode lengths from 3D point clouds of cucumber plants. First, a deep neural network based on PointNet++ was trained to segment the point clouds into plant parts. The points that were predicted as 'node' were then selected and a clustering algorithm was used to group points belonging to the same node. The Euclidean distance between the detected nodes was used as an estimate of the internode length. The results were compared to the results of a previously published method based on 2D images.

The results of the 3D method were significantly more accurate than the results of the 2D method. Moreover, in contrast to the 2D method, the internode length estimates of the 3D method were equally accurate for curved plants as well as for straight plants. The results clearly demonstrated that computer-vision based methods to measure plant architecture in general, and more specifically to measure internode length, greatly benefit from the availability of 3D data.

5.1 Introduction

Plants grown in a commercial setting are subject to requirements set by plant growers, retailers, and consumers. For example, plants should have a high resistance against diseases and pests, produce tasty fruits and have a high yield at the right time. These characteristics of the plant are determined by the genetics of the plant, and by the environment in which the plant is growing, including crop management activities like de-leafing or harvesting. Plant breeding companies are developing new varieties, that are able to meet the set requirements.

For an efficient plant breeding process, breeders need to unravel the interaction between genotype, phenotype, and environment. Studying the genotype-phenotype-environment interaction requires a detailed registration of all three of these components. In this work, the focus is on measuring the phenotype of the plant. Traditionally, measuring the phenotype of the plant, called phenotyping, is done based on human observation. This is a time-consuming process (Gehan & Kellogg, 2017) and the obtained data is often subjective. In contrast, automated digital phenotyping methods open up opportunities to obtain large-scale phenotypic data in an objective way (Costa et al., 2019; Tripodi et al., 2022). Simple traits can be measured from 2-dimensional (2D) images, but measuring more complex traits, like an accurate estimation of internode length, leaf size or branching angle, requires using 3-dimensional (3D) data (Boogaard et al., 2020; Minervini et al., 2015).

The 3-dimensional organisation of plant parts, known as plant architecture (Reinhardt & Kuhlemeier, 2002), is defined by a set of variables like plant height, leaf or branching angle, leaf size, and internode length. Plant architecture has an effect on the amount of intercepted light and consequently also on photosynthesis and plant productivity. Furthermore, changes in plant architecture can indicate plant stress. Changes in internode length for example, have been related to stress caused by drought and salinity (Litvin et al., 2016; Najla et al., 2009; Sibomana et al., 2013). Early observation and quantification of the stress response helps to breed for more tolerant varieties. Therefore, we have taken measurement of internode lengths as a use case.

In earlier work, an automated method to measure internode length in cucumber plants was presented (Boogaard et al., 2020). This method was based on 2-dimensional images. These images were taken from multiple viewpoints around the plant. In each image, the nodes were detected and the detections from the different viewpoints were then combined. Based on the combined detections, the

Euclidean distance between the nodes was taken as an estimate for the internode length. For the conversion from pixels to mm, a fixed plant-camera distance was assumed. However, it turned out that for 25% of the plants, the curves of the stem led to a significant violation of this assumption. The reported mean absolute error of the estimated internode lengths for plants that had a straight growing pattern was 7.7 mm, while the mean absolute error for the curved plants was 23.0 mm. This suggested that accurate internode-length estimation could only be achieved when 3D data was used, as in that case, the fixed plant-camera distance would no longer be required.

Therefore, in the current work, we hypothesise that "*using 3D point clouds instead of 2D images allows to estimate internode lengths of curved plants with the same accuracy as obtained for straight plants"*, and furthermore, that *"the error of the internode lengths estimated from the 3D point clouds is smaller than the error of the internode lengths estimated from the 2D images"*.

Contributions of the paper

To test the two hypotheses, we extended the 3D point-cloud segmentation method of Boogaard, Van Henten, and Kootstra (2021, 2022). This method was developed to segment point clouds of plants into plant parts. Based on the segmented point cloud, we first propose a node instance detection algorithm, to identify how the segmented node points could be grouped, resulting in a set of node objects.

The internode length was then determined as the distance between two consecutive nodes along the stem of the plant. To this end, for each node it was determined whether the next node of the plant was also detected. If that was the case, the corresponding internode length was estimated as the Euclidean distance between the two nodes in 3D space, similar to how the internode length was estimated in the 2D method in 2D space.

The first hypothesis was tested by comparing the 3D internode lengths for straight and curved plants. For the second hypothesis, the internode lengths of the same plants were also estimated based on the 2D method of Boogaard, Rongen, and Kootstra (2020). The 2D internode length estimates were then compared to the 3D internode length estimates.

In the remainder of this paper, first, the materials and methods are presented in section 5.2, starting with an overview of the dataset in section 5.2.1. The 2D method is summarized in section 5.2.2 and the 3D method is presented in section 5.2.3. The evaluation methods are then introduced in section 5.2.4. The results of the point-cloud segmentation, clustering algorithm, node detection, and internode length estimation are presented in section 5.3. Finally, the results are discussed in section 5.4 and in the conclusion in section 5.5, we reflect on the hypotheses that were presented above.

5.2 Materials & Methods

In this section, the materials and methods are presented, starting with a description of the data acquisition and the resulting 2D and 3D dataset in section 5.2.1. The method to estimate internode lengths from the 2D images is summarized in section 5.2.2. The approach to estimate the internode lengths from the 3D point clouds is explained in section 5.2.3. Finally, in section 5.2.4, the evaluation methods are presented.

5.2.1 Data

Twelve cucumber plants (Proloog RZ F1, Rijk Zwaan, De Lier, The Netherlands) were placed on plant gutters with a plant distance of 1 meter to prevent occlusion between plants. A schematic overview of the plants is provided in Figure 5.1.

The data acquisition was done using a Wiwam plant analyser (WIWAM, 2022) equipped with a 2D camera used for the 2D internode length estimation and a Phenospex F500 dual scan system (Phenospex, 2017) for the 3D internode length estimation. The 2D camera that was used is an IMPERX B4820 16 MP CCD camera with an image resolution of 4904 x 3280 pixels (IMPERX, 2018). The 3D data was based on laser-line triangulation and was collected using a Phenospex PlantEye F500. This is a multispectral 3D scanner for plant phenotyping, which also provided spectral information (red, green, blue and NIR) for each point in the point clouds.

The phenotyping system was able to move to different positions, or viewpoints, around each plant, see Figure 5.1 for a schematic overview. At each of these viewpoints, the mobile platform stopped and the sensor head moved in a vertical direction, starting at the plant gutter. During the upwards movement, the Phenospex F500 dual scan system generated a point cloud of the plant. The 2D images of the same plant were then collected during the downwards movement. The 2D images were collected at six height levels, from all six (I-VI) viewpoints around the plant. The 3D point clouds were collected only from viewpoints II and V, at both sides of the plant gutter, straight in front of the plant. However, since the dual scan system consisted of two scanners, each scan provided 2 viewpoints of the plant. Having multiple viewpoints reduced the number of occluded nodes. The effect of a different number of viewpoints for the 2D dataset as compared to the 3D dataset is discussed in section 5.4.1.

Data collection was done in 2018, between June 25th and July 5th. The 2D images were taken multiple times a day, resulting in a dataset of 9,990 images. The 3D point clouds of the same plants were collected only once per day, resulting in 264 point clouds. As a reference, the internode length of all plants was measured with a measuring tape. This was done on the third, fifth and eighth day of the experiment. The internode length measurements ranged from 10 mm to 175 mm. The distribution of the internode lengths that were measured on the three days is shown in Figure 5.2. The total number of internode length measurements obtained was 357. In (Boogaard et al., 2020), the mean absolute error in these measurements was estimated to be 2.7 mm, based on a comparison of multiple measurements of the same internode.

The number of nodes in a plant was always one more than the number of internodes of that plant. As the 12 plants were measured on 3 days, the total number of nodes present in the scene was 357 internodes + (12*3) = 393 nodes.

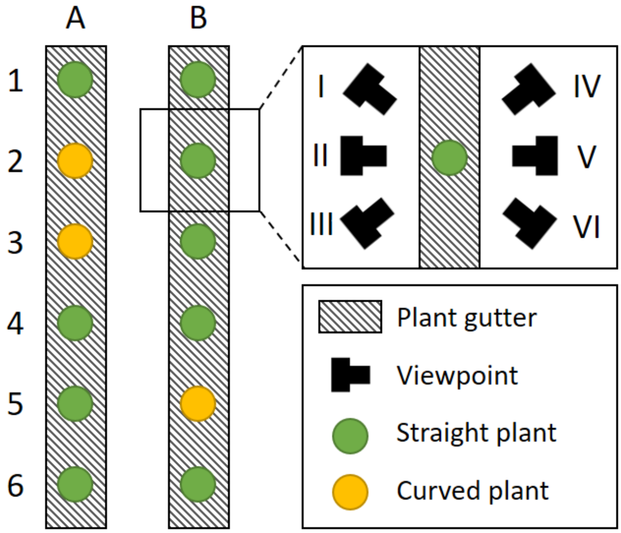


Figure 5.1 – Schematic top view of the experimental set-up. The plants were grown on two plant gutters (A and B), containing six plants (1-6) each. The curved plants A2, A3 and B5 were referred to as outlier plants 4, 5 and 8 in (Boogaard et al., 2020). The six viewpoints (I-VI) are shown in the zoomed in part at the top right corner of the figure.

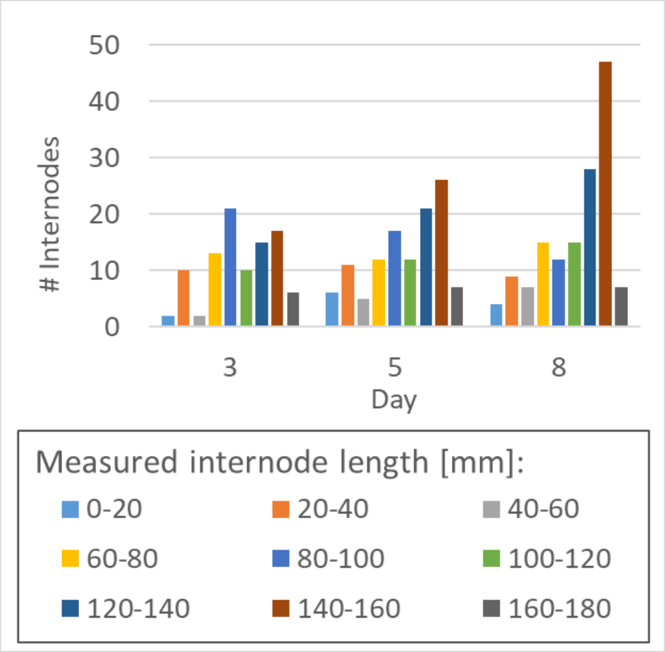


Figure 5.2 – Histogram showing the distribution of internode lengths measured with the measuring tape on days 3, 5, and 8 of the experiment, for all plants. The bars indicate the number of internodes that fall within the bins defined in the legend of the figure.

Each plant was attached to a supporting wire to enforce vertical plant growth. However, as mentioned in the introduction, several plants still showed a curved growing pattern, making it infeasible to accurately estimate internode lengths for these plants using the 2D method. The plants for which this was the case (A2, A3 and B5) are marked in yellow in Figure 5.1. These plants were identified based on visual observation by Boogaard, Rongen, and Kootstra (2020). The 3D data used in this work allowed to quantify the difference between straight and curved plants. For this purpose, the Root Mean Square Error (RMSE) was calculated, using the difference between the actual plant-camera distance and the assumed plant-camera distance as the error. The RMSE was calculated for all nodes on the last day of the experiment, as larger plants showed more curvatures due to the weight of the plant. In Figure 5.3 it can be seen that, indeed, the plants that were identified by Boogaard et al. (2020) showed a higher RMSE, indicating that the plant-camera distance was not as assumed for these plants.

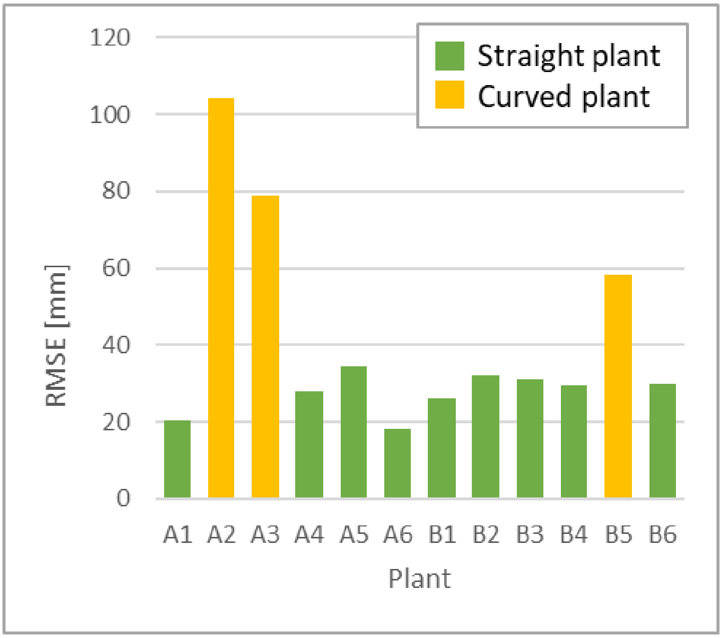


Figure 5.3 – RMSE of the plant-camera distance for all plant nodes, on day 7. The curved plants A2, A3, and B5 clearly have a larger deviation from the assumed plant-camera distance than the straight plants.

5.2.2 2D internode length estimation

The 2D internode lengths were estimated using the method presented by Boogaard, Rongen, and Kootstra (2020). For completeness, a summary of the method is added in this section. For more details, we refer to the mentioned paper.

The set of 2D images was used to estimate the 2D internode lengths. First, the nodes were detected in the individual images using a deep-learning based object-detection algorithm, based on YOLO-v3 (Redmon & Farhadi, 2018). Using images in which the nodes were manually labelled, the network learned to predict the location of nodes in new images. This resulted in a list of node locations per image. The node locations per image were then transformed to world coordinates, resulting in a list of detected nodes per plant.

Often, nodes were detected in multiple viewpoints. Multiple detections of the same node were combined using a clustering algorithm called affinity propagation (Frey & Dueck, 2007). The goal of the clustering algorithm was to obtain a list of node locations in which each node was present only once. The clustered list of nodes was then sorted according to location on the stem, starting with the lowest node as number 1 and counting upwards along the stem of the plant. The internode length was estimated as the Euclidean distance between two consecutive nodes. A simple linear regression model was then fitted to correct for a systematic error that was found in the estimated internode lengths.

5.2.3 3D internode length estimation

In this section, the method to estimate internode lengths from the 3D point clouds is presented in three steps: i) segmentation of the point clouds into plant parts, ii) detection of nodes from the segmented point clouds, and iii) estimation of the internode lengths.

5.2.3.1 Point-cloud segmentation

The segmentation of the 3D point clouds into plant parts was based on the deep neural network PointNet++ (Qi et al., 2017), following the implementation presented by Boogaard, Van Henten, and Kootstra (2021). The neural network combines features on a per point basis, with contextual features that represent the neighbourhood of the point. To learn these contextual features, a set of points was selected from the entire point cloud. These points were chosen such that the distance between the points was maximised, meaning that all regions of the point cloud were covered. For each of these points, the neighbourhood points were then selected as a group. For each group of points, a PointNet layer was used to learn local features within the group. Multiple of these set abstraction layers with a range of receptive fields were applied, to learn features from local to global scale. As output, the network labels every point to belong to one of the plant parts.

To train the network to learn features relevant for plant-part recognition, a training dataset was required. Therefore, the 264 point clouds were manually segmented in stem, petiole, leaf, growing point, node, ovary, tendril, and non-plant (e.g. the plant gutter, supporting wire and stick). This was done using CloudCompare (CloudCompare, 2019). An example of a manually segmented point cloud obtained on the first and last measurement day is shown in Figure 5.4.

As can be seen in Figure 5.4, not all classes were equally present in the data. To improve the segmentation of the underrepresented classes, the approach presented in Boogaard, Van Henten, and Kootstra (2022) was followed. Due to computational limitations, it is common to divide the point cloud into smaller chunks, which are then independently processed by the network and recombined later to obtain a segmentation of the full point cloud. By sampling more chunks of underrepresented classes, the level of class balance in the training data was improved. In this way, the segmentation quality of the underrepresented classes, such as the node, significantly improved. In this work, we applied the class-dependent sampling strategy, using the default chunk size of 4,096 points.

The labelled data was split in a training, validation, and test set. This was done on plant level, to prevent that point clouds of the same plant were added to multiple sets. The training set consisted of 10 plants (220 point clouds) and the validation and test set each contained the 22 point clouds of one of the remaining plants. The network was trained twelve times, each time having a different plant in the validation and test set. The validation set was used to determine when to stop training. The weights obtained at the minimum validation loss were then used to segment the point clouds in the test set. By repeating this for all twelve test sets, a predicted segmentation of all point clouds was obtained.

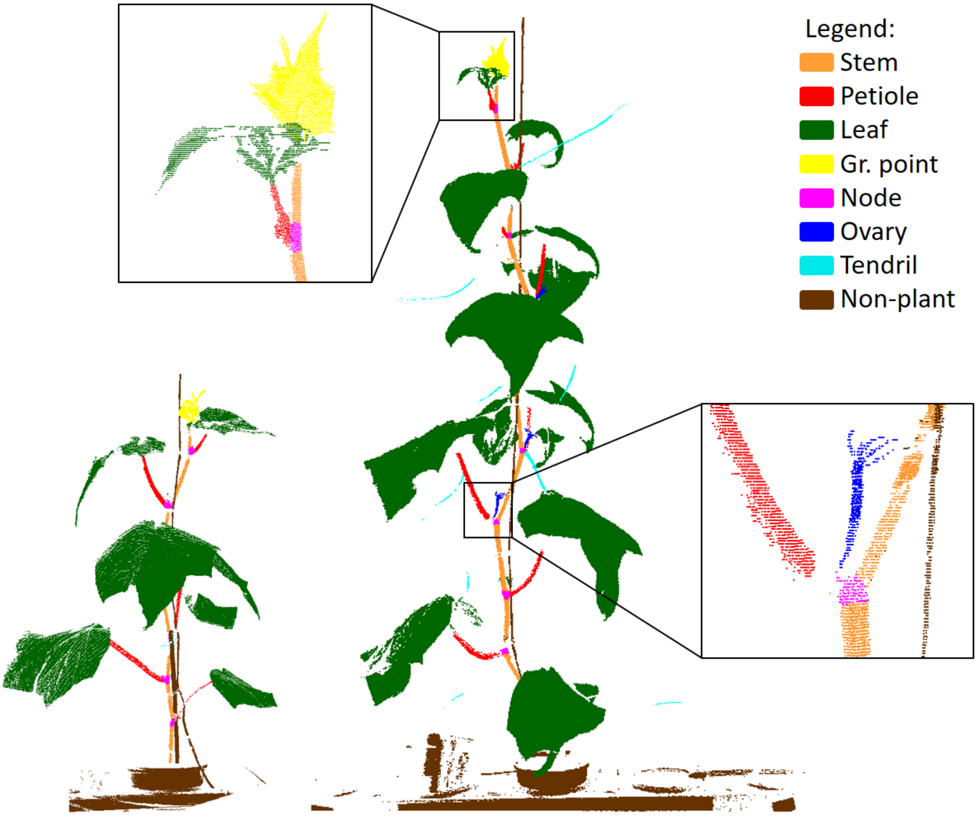


Figure 5.4 – Example of a manually annotated point cloud on the first (left) and the last (right) day of data acquisition. The colours represent the classes as specified in the legend, the black squares show a zoomed in segment of the point cloud. Adapted from (Boogaard et al., 2022).

5.2.3.2 Node detection

To reduce the level of occlusion, the point clouds obtained from viewpoints II and V, from both sides of the plant gutter, were combined. To this end, the point clouds had to be transformed from the camera coordinate system to the world coordinate system. First, the coordinate system of viewpoint II was set as the world coordinate system. Then, the point cloud obtained from viewpoint V was manually transformed in CloudCompare (CloudCompare, 2019). The origin of the point cloud was defined at the centre of the pot. A rotation of 180° around the vertical axis was applied, followed by small corrections to increase the accuracy of the registration. This procedure was done once for each plant position at the beginning of the measurement campaign. In future work, this procedure can be automated by adding a few markers to the scene. The resulting transformation matrix was stored and applied to transform all point clouds from that plant position. From the combined point clouds, all points that were predicted to be 'node' were selected.

To determine which of the selected points belonged to the same node, the node points were clustered using the HDBSCAN clustering algorithm (Campello et al., 2013). HDBSCAN is a density-based clustering algorithm that is particularly suited for finding clusters of different size and shape, in datasets that contain noise or outliers. It is not required to specify the number of clusters beforehand. The only input variable required is the minimum number of points in a cluster, called the minimum cluster size. To provide insight in the balance between number of missed nodes (false negative) and number of wrongly identified nodes (false positives), the precision, recall, and F1-score of the node detection algorithm for a range of values for the minimum cluster size are reported in section 5.3.2. The setting that resulted in the highest F1-score was used to obtain the node detections for the remainder of this research. The centre point of each detected node was taken as an estimate for the location of that node.

5.2.3.3 Internode length estimation

The result of the node detection step was a list of estimated node locations for each plant on each day. The node locations were sorted by increasing height, such that the first location in the list corresponded to the node closest to the pot and the last node of the list corresponded to the youngest node in the top of the plant. The internode length was then estimated as the Euclidean distance between two consecutive node locations.

Similar to the 2D case, the estimated internode lengths showed a systematic error. The specific reason for this error was not fully investigated, but possible explanations include errors in the calibration of the measurement set-up, inaccuracies in the placement of the plants and the plant gutter, and measurement errors in the collected reference measurements. For a fair comparison, the same procedure as was used for the 2D internode length estimation was used to correct for the systematic error in the 3D internode lengths. For each test plant, a simple linear regression model was fit on all other plants. That model was then used to calibrate the estimated internode lengths of the test plant.

5.2.4 Evaluation methods

In this section, the evaluation methods for the point-cloud segmentation, node detection, and internode-length estimation are presented.

5.2.4.1 Point-cloud segmentation

The evaluation of the point-cloud segmentation method was based on a comparison of the manually assigned labels and the predicted labels. A confusion matrix was used to provide insight into what classes were confused with other classes.

Furthermore, the Intersection-over-Union (IoU) for each class is presented. This metric is based on a point-wise comparison between the predicted segmentation and the manually obtained segmentation. If a point was assigned to the same class in both cases, it was considered a true positive (TP). Points that were not assigned to the same class, were considered false negatives (FN) for the manually assigned class and false positives (FP) for the predicted class. For each class c, the IoU was then calculated according to Equation 5.1.

$$\mathrm{IoU_c} = \frac{\mathrm{TP_c}}{\mathrm{TP_c} + \mathrm{FP_c} + \mathrm{FN}_{\mathrm{c}}} \qquad [\text{-}] \qquad \textit{Equation 5.1}$$

5.2.4.2 Node detection

To evaluate the detected nodes, first, the ground truth location of each node was required. All nodes that were present in the plants were given a unique ID based on the plant number, the day of measurement, and the order of the node on the stem. The nodes that were identified in the manually labelled data were then linked to the corresponding real nodes. For each manually labelled node, the centre was stored as the xyz-location of that node.

The estimated node locations from the point clouds that were segmented by the segmentation algorithm were then compared to the node locations obtained from the manually labelled data. For each of the nodes identified in the manually labelled data, the Euclidean distance to all estimated node locations was calculated. The estimated node location with the lowest distance was linked to the manually detected node and removed from the list, to make sure it was not linked to another node. If no estimated node location was found within 20 mm of the manually detected node, the node was considered not to be detected by the algorithm.

The manually detected nodes that were linked to an estimated node location were considered to be true positives (TP). The undetected nodes were considered false negatives (FN) and the estimated node locations for which no corresponding manually detected node was found were considered false positives (FP). From the number of TP, FN, and FP, the precision (P), recall (R), and F1-score (F1) of the node detection algorithm was calculated according to Equations 5.2-5.4.

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \qquad [\text{-}] \qquad \textit{Equation 5.2}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \qquad [\text{-}] \qquad \textit{Equation 5.3}$$

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{P} \cdot \mathrm{R}}{\mathrm{P} + \mathrm{R}} \qquad [\text{-}] \qquad \textit{Equation 5.4}$$

5.2.4.3 Internode length estimation

The internode length measurements obtained with a measuring tape were used to evaluate the accuracy of the estimated internode lengths. Since the reference measurements were taken on the third, fifth and eighth day of the experiment, we also used the estimated internode lengths for these days. Furthermore, if one or two nodes of an internode were not detected, the corresponding internode length could not be estimated and those cases were not taken into account in this evaluation. This is further discussed in section 5.4.3. The number of estimated internode lengths that was used for the evaluation is reported in section 5.3.4.

The difference between the estimated internode length, $\hat{s}_i$, and the manually measured internode length, s_i, was called the error e_i, for the internode length between node i and node $i+1$, according to Equation 5.5.

$$e_i = \hat{s}_i - s_i \qquad [\text{mm}] \qquad \textit{Equation 5.5}$$

To test the significance of differences in the errors between the 2D internode lengths and the 3D internode lengths, and between the curved plants and the straight plants, a Mann-Whitney U test was used.

5.3 Results

The results of this paper are presented in this section. The segmentation performance is presented in section 5.3.1, showing how well the plant parts were identified in the point clouds. In section 5.3.2, the results of the clustering algorithm are given for different values of the minimum cluster size. The visibility and detection of the nodes is then presented in section 5.3.3. Finally, in section 5.3.4, the results of the estimated internode lengths are presented and compared to the internode lengths obtained from the 2D data. Intermediate results from the 2D method are not presented, for these results we refer to (Boogaard et al., 2020).

5.3.1 Point-cloud segmentation

The point clouds of the twelve test sets were segmented by the corresponding twelve trained models. The predicted labels were then compared to the manually assigned labels. The resulting confusion matrix is shown in Table 5.1. Each row corresponds to the points that were manually assigned to that class and shows how these points were labelled by the network. The correctly predicted points are shown on the diagonal. The best performance was obtained for the classes ‘leaf’ (98.8%) and ‘non-plant’ (98.1%). Performance decreased for ‘stem’, ‘growing point’, ‘petiole’, ‘tendril’, and ‘ovary’. The lowest percentage of correctly predicted points was observed for the class ‘node’ (43.0%). The majority of incorrectly classified ‘node’ points was predicted to be ‘stem’.

The mean IoU-values per class are presented in Figure 5.5, including the 95%-confidence interval on the mean. Again, the classes ‘leaf’ and ‘non-plant’ showed the best performance, while the lowest mean IoU-value was observed for the ‘node’. The small confidence intervals indicate a consistent performance over the point clouds in the test sets.

Table 5.1 – Confusion matrix of the trained networks. Each row corresponds to the points that were manually labelled as that class, showing how the network predicted those points. The values on the diagonal are correct predictions and correspond to the recall. The correct predictions are highlighted in green and the wrong predictions are highlighted in red, where brighter colours correspond to higher values.

Confusion matrix (percentages)		Predicted labels							
		Stem	Petiole	Leaf	Gr. point	Node	Ovary	Tendril	Non-plant
Manually assigned labels	Stem	**90.2**	2.1	1.3	1.1	2.9	0.3	0.4	1.6
	Petiole	6.9	**82.5**	3.9	1.0	2.8	1.6	0.9	0.5
	Leaf	0.1	0.1	**98.8**	0.6	0.0	0.0	0.1	0.3
	Gr. point	1.6	0.3	8.1	**88.4**	0.0	0.0	0.5	1.2
	Node	44.0	7.0	1.2	1.2	**43.0**	1.1	0.8	1.7
	Ovary	9.0	11.9	2.9	0.3	4.9	**67.6**	1.8	1.5
	Tendril	6.4	3.2	8.8	1.2	1.5	1.2	**72.6**	5.1
	Non-plant	0.9	0.1	0.4	0.1	0.0	0.0	0.4	**98.1**

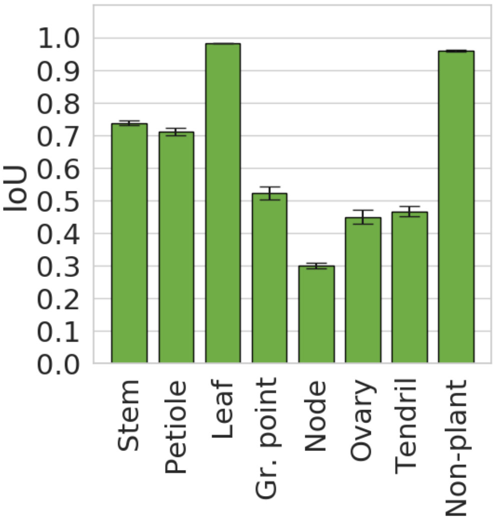


Figure 5.5 – IoU-values based on the predicted segmentation of the test sets. The bars indicate the mean IoU-values and the error bars show the 95% confidence interval on the mean.

5.3.2 Clustering algorithm

The points that were predicted to be 'node' were clustered by the HDBSCAN-algorithm. The effect of different values for the input parameter 'minimum cluster size' was tested and the results are shown in Figure 5.6. The highest precision (0.93) was found for a minimum cluster size of 50 points. For larger values, the number of correctly detected nodes dropped, since smaller nodes were either discarded or merged with other nodes in their neighbourhood. For smaller minimum cluster sizes, the number of correctly detected nodes kept increasing, however, the precision was lower due to a higher number of false positive detections. These false positive detections do not have an effect on recall, and therefore, the highest recall (0.85) was found for a minimum cluster size of 2 points. The decrease in recall for larger values of the minimum cluster size is due to an increasing number of false negatives.

The maximum value of the F1-score (0.87) was observed for a minimum cluster size of 20 points. Therefore, this setting was used for the remainder of this research.

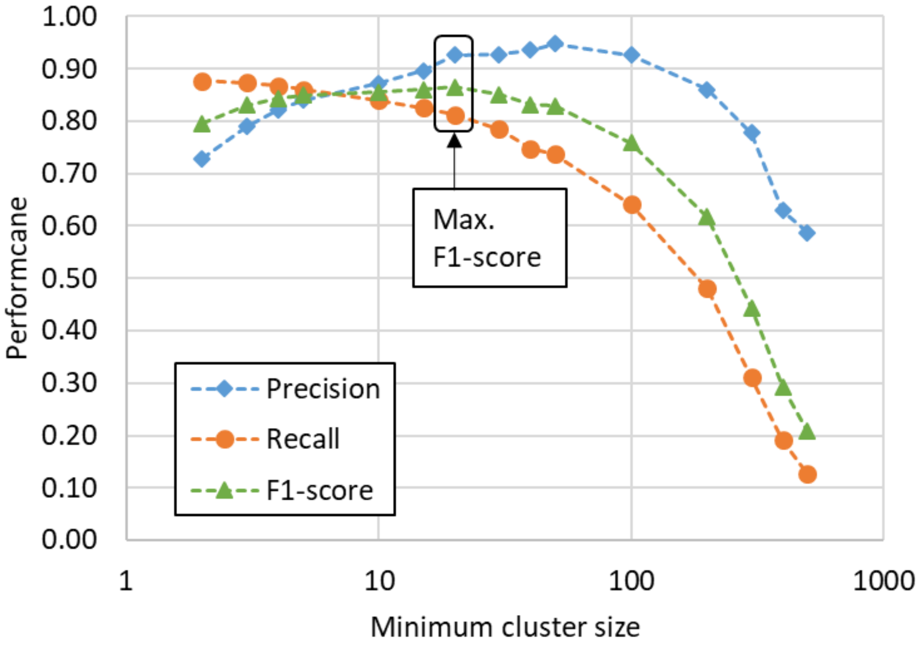


Figure 5.6 – Precision, recall and F1-score of the clustering algorithm for different values of the minimum cluster size used by the HDBSCAN-algorithm. Note that the x-axis, showing the minimum cluster size, has a logarithmic scale for readability of the figure. The selected value used in the remainder of this research (minimum cluster size = 20) is indicated in the figure.

5.3.3 Node detection

As mentioned in section 5.2.1, the total number of nodes was 393. However, not all of the nodes that were present in the plants, were also found in the data. Based on the manually labelled point clouds, the number of nodes that was identified by the annotator was 293. This means that 100 of the real nodes, were not identified by the annotator in the collected point clouds. Further analysis of these missing nodes showed that 54 of them concerned the two youngest nodes of the plant. During manual annotation of the point clouds, the top of the plant was labelled as 'growing point'. It turned out that this class overlaps with the youngest nodes, which were therefore not labelled and also not learned by the segmentation algorithm. The remaining 46 nodes that were not identified in the point clouds were mostly occluded by leaves.

Based on the node points that were predicted by the segmentation algorithm, the clustering algorithm found 257 clusters of node points. After linking these clusters to the manually annotated nodes, 238 clusters were identified as correctly predicted nodes (true positive), meaning that the remaining 19 clusters were false positives. Furthermore, this means that 293 – 238 = 55 nodes were not detected, or false negatives. Based on this data, the precision of the node detection algorithm was 0.93 (238/257) and the recall was 0.81 (238/293), leading to an F1-score of 0.87. In comparison, the precision and recall of the node detection step for the 2D method were 0.95 and 0.92 respectively, leading to an F1-score of 0.93 (Boogaard et al., 2020).

The results of the node detection are schematically summarised in Figure 5.7. Note that, even while only 43.0% of the 'node' points in the point clouds were correctly segmented (see Table 5.1), as long as part of the points of a node were found, the clustering algorithm could detect this node.

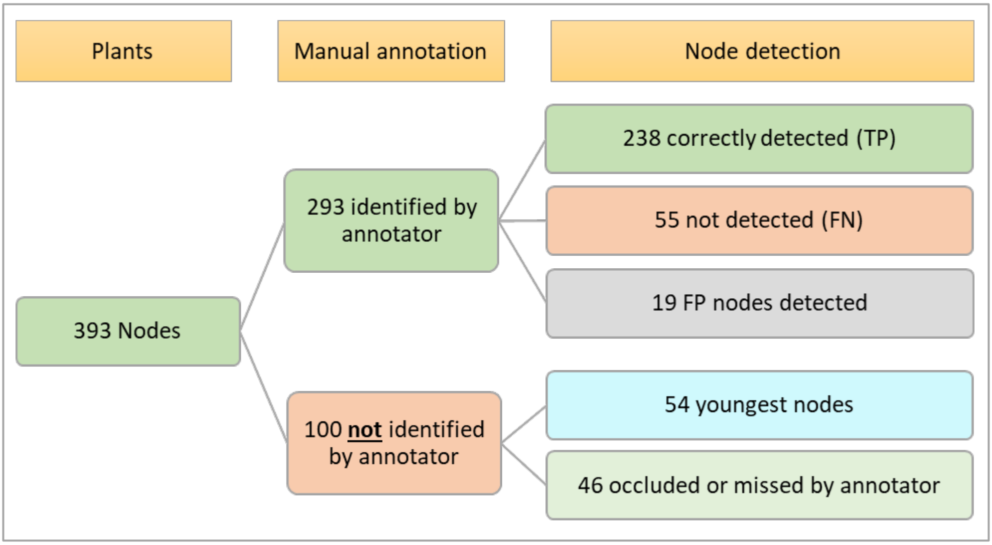

Plants
Manual annotation
Node detection
393 Nodes
293 identified by annotator
100 not identified by annotator
238 correctly detected (TP)
55 not detected (FN)
19 FP nodes detected
54 youngest nodes
46 occluded or missed by annotator

Figure 5.7 – Classification of the nodes that were present in the plants, divided over identified and not identified by the annotator of the point clouds. For the identified nodes, the number of detected nodes for the 3D method is reported.

5.3.4 Internode length estimation

The estimated internode lengths were evaluated for internodes for which both nodes were correctly detected, as explained in section 5.2.4.3. The number of internodes that was present in the plants, based on the manual measurements, was 357. The 2D method was able to correctly detect both nodes for 346 of these internodes. In the 3D point clouds, the annotator identified both nodes for 230 internodes. The 3D node detection algorithm detected both nodes for 171 internodes.

The manually measured internode lengths were plotted against the estimated internode lengths from the 2D images and from the 3D point clouds, as shown in Figure 5.8. First, what is clearly visible, is that the 3D internode lengths were more accurate than the 2D internode lengths. Furthermore, there was no visible difference between the straight plants and the curved plants in the results of the 3D method, while the curved plants clearly have a larger error in the results of the 2D method.

In the lower-left corner of the two plots, it can be seen that the 2D method was able to estimate shorter internode lengths than the 3D method. As mentioned in section 5.2.1, the shortest internode length that was measured with the measuring tape was 10 mm. However, the shortest internode for which the 3D method detected both nodes and thus could estimate the internode length, was manually measured to be 45 mm. So, despite the higher accuracy of the 3D method, short internodes were not measured. The main reason for this is that the nodes in the top of the plant, corresponding to the shorter internodes, were overlapping with the class ‘growing point’, as explained in section 5.3.3.

A boxplot summarizing the absolute errors of the internode length estimates, for straight and curved plants, and for 2D images and 3D point clouds is shown in Figure 5.9. To test if the internode lengths estimated for the curved plants were significantly different from the internode lengths estimated for the straight plants, a Mann-Whitney U test was used. For the 2D images, the null hypothesis was rejected (p<0.001), indicating that internode lengths estimated by the 2D method for the straight plants were significantly more accurate than for the curved plants. Also for the 3D method, the absolute errors for the straight plants were compared to the absolute errors for the curved plants. In this case, the null hypothesis was accepted (p=0.29), meaning that the absolute errors for the straight plants were not significantly different from the errors for the curved plants. In other words, indeed, the 3D method is able to measure the curved plants with the same accuracy as the straight plants.

Besides the comparison between curved and straight plants, we tested if the internode lengths estimated from the 3D point clouds were more accurate than the internode lengths from the 2D images, as Figure 5.8 suggested. It was found that the absolute errors of the 3D method for straight plants (median: 4.1 mm, mean: 5.2 mm) were significantly lower (p<0.001) than the absolute errors of the 2D method for straight plants (median: 6.33 mm, mean: 7.7 mm). Furthermore, also the absolute errors of the 3D method for curved plants (median: 4.0 mm, mean: 4.7 mm) were found to be significantly lower (p<0.001) than the absolute errors of the 2D method for straight plants.

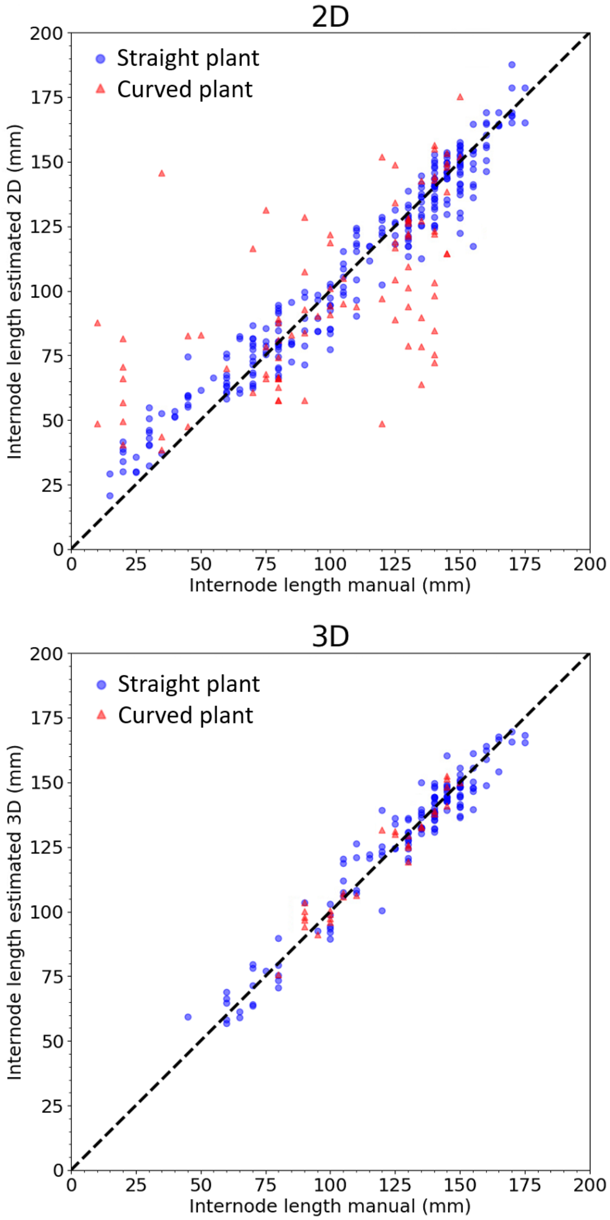


Figure 5.8 – The manually measured internode length (x-axis) plotted against the estimated internode length (y-axis). The upper plot shows the results based on the 2D method, the lower plot shows the results of the current work based on the 3D point clouds. The nodes of the straight plants are shown as blue circles, while the nodes of the curved plants are shown as red triangles. The dashed line shows the function x=y, as an indication of where the estimated internode length was equal to the reference measurement.

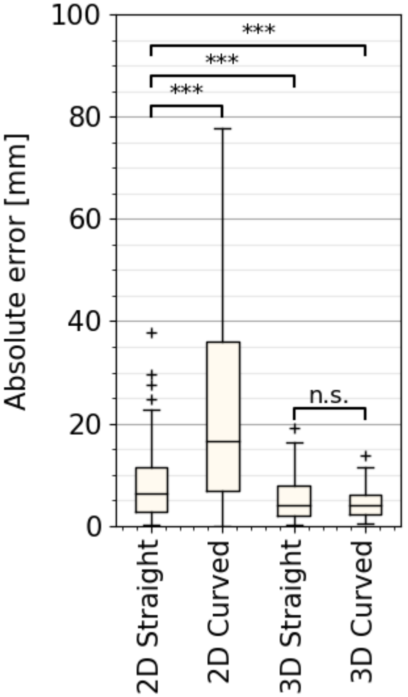


*Figure 5.9 – Absolute error (mm) of the predicted internode lengths based on 2D and 3D data, for straight and curved plants. The boxes show the lower and upper quartile, the whiskers indicate the highest and lowest absolute error within 1.5 times the inter-quartile range. Values outside this range were considered outliers and are shown as '+'. The level of significance above the arches is indicated as 'n.s.', meaning no significant difference was observed, or as '***', meaning a significant difference was observed with p<0.001.*

5.4 Discussion & recommendations

The results presented in this paper clearly show that the internode lengths estimated from the 3D point clouds were more accurate than the internode lengths estimated from the 2D images. However, as mentioned in section 5.3.4, the number of detected nodes and thus the number of internode lengths that could be estimated was higher for the 2D method. In this section, these results will be further discussed, starting with the node visibility in section 5.4.1. The node detection is discussed in section 5.4.2. The estimated internode lengths are discussed in section 5.4.3, and the final recommendations are given in section 5.4.4.

5.4.1 Node visibility

The level of occlusion, or the total number of nodes that was present in the acquired data, depended on the number of viewpoints used. For the 2D images, it was shown that the percentage of visible nodes increased from 91.9% to 99.3%, when all six viewpoints around the plant were used instead of only viewpoint II and V (Boogaard et al., 2020). In comparison, the percentage of visible nodes in the 3D point clouds was 74.6% (293/393). Out of the 100 nodes that were not visible, 46 nodes were occluded by leaves or other plant parts. Based on the increased visibility rate observed in the 2D data, it is likely that also in the 3D data, more nodes would be visible when adding more viewpoints. An alternative would be to use an active vision approach (Burusa et al., 2022; Wu et al., 2019). In such a system, the camera could be mounted on a robot arm, providing more flexibility to use additional viewpoints. In combination with an online node recognition method, the system could be optimised to add new viewpoints until as many nodes as possible are included in the data.

The remaining 54 nodes that were not identified by the annotator were labelled as ‘growing point’. The height of the annotated growing point was in the range of 50-70 mm, an example can be seen in Figure 5.4. When collecting the reference measurements, nodes were measured if they were longer than 10 mm, meaning that often the 1 or 2 youngest nodes were hidden inside the growing point. However, the resolution of the point clouds was not sufficient to identify these nodes in the point cloud. It could be tested if other sensors or a combination with the 2D images, which had a higher resolution, could improve the detection of the youngest nodes. However, this also leads to the question whether it is relevant to measure internode length in such an early stage. This goes beyond the scope of the current work, but the specific requirements of automated internode length measurements should be discussed in more detail with plant scientists in future work. The results presented in this paper contribute to this discussion, by providing benchmarks of what can be expected when using different approaches.

5.4.2 Node detection

Besides the higher visibility rate of nodes in the 2D dataset, also the detection rate was higher. The F1-score for node detection from the 2D images was 0.93, while the F1-score for the detection of nodes from the 3D point clouds was 0.87. Both the number of false positives as well as the number of false negatives has a negative effect on the F1-score, which will be discussed in the next two paragraphs.

The total number of false positive node detections in the 3D data was 19. Further inspection showed that 15 of the false positives could be explained due to other plant parts that were predicted as nodes. In the remaining 4 cases, the node points of two nodes that were close to each other, were combined into one cluster. The centre of such a cluster, which was used as the estimated node location, was in the middle between the two nodes. Therefore, the distance to either of these two nodes was too large, such that the cluster was not linked to the manually detected nodes and was considered a false positive. Also, since this link was not made, the two manually detected nodes remained undetected and were both considered a false negative.

The total number of false negative detections in the 3D data was 55. These nodes were mainly undetected because the node points were predicted to be stem or petiole, as can be seen in Table 5.1. The low IoU for the class ‘node’ (0.3) also indicates that the segmentation quality could be improved. One approach would be to increase the amount of training data further, aiming to better represent the variation in the data. However, obtaining high quality training datasets for point cloud segmentation, especially in the plant domain, is still challenging (Boogaard et al., 2021; Dutagaci et al., 2020; Turgut et al., 2022). Alternatively, attempts to focus on node detection only, instead of having a full segmentation of the point cloud, might improve the detection rate. However, when looking at the broader perspective of measuring plant architecture, a properly segmented point cloud is a valuable asset, since it opens up possibilities to measure other plant traits.

5.4.3 Internode length estimation

In the results section, we showed that the proposed method to estimate 3D internode length improved the internode length estimates for curved plants and outperformed the previous 2D method in terms of accuracy also for the straight plants. However, this is based only on the internodes for which the two corresponding nodes were detected. If a specific node was not detected, the internode length below as well as the internode length above the undetected node could not be estimated.

In the current paper, both the evaluation of whether a detected node was correct, as well as the check whether two nodes were belonging to the same internode, was done based on the manually annotated data. While this is a valid approach to answer the research questions that were dealt with in this paper, it hampers the practical applicability of the method. Therefore, besides improving the node detection step, further research into automated detection of missing nodes is recommended. One approach could be to use the segmented stem elements to determine if two detected nodes are connected.

5.4.4 Final recommendations

Both the 2D and the 3D method have their advantages and disadvantages. One of the main limitations of the 2D method was that not all plants could be accurately measured, especially if the plants showed a large variation in plant-camera distance. For the 3D method, one of the main limitations was that not all nodes were visible and detected, meaning that not all internodes lengths could be estimated. To overcome the limitations of the 2D and the 3D method, we recommend to combine both datasets. It is expected that a combined approach that uses the high resolution of the 2D images for accurate detection and the spatial information contained in the 3D point clouds to correct for variations in plant-camera distance can solve the limitations indicated above.

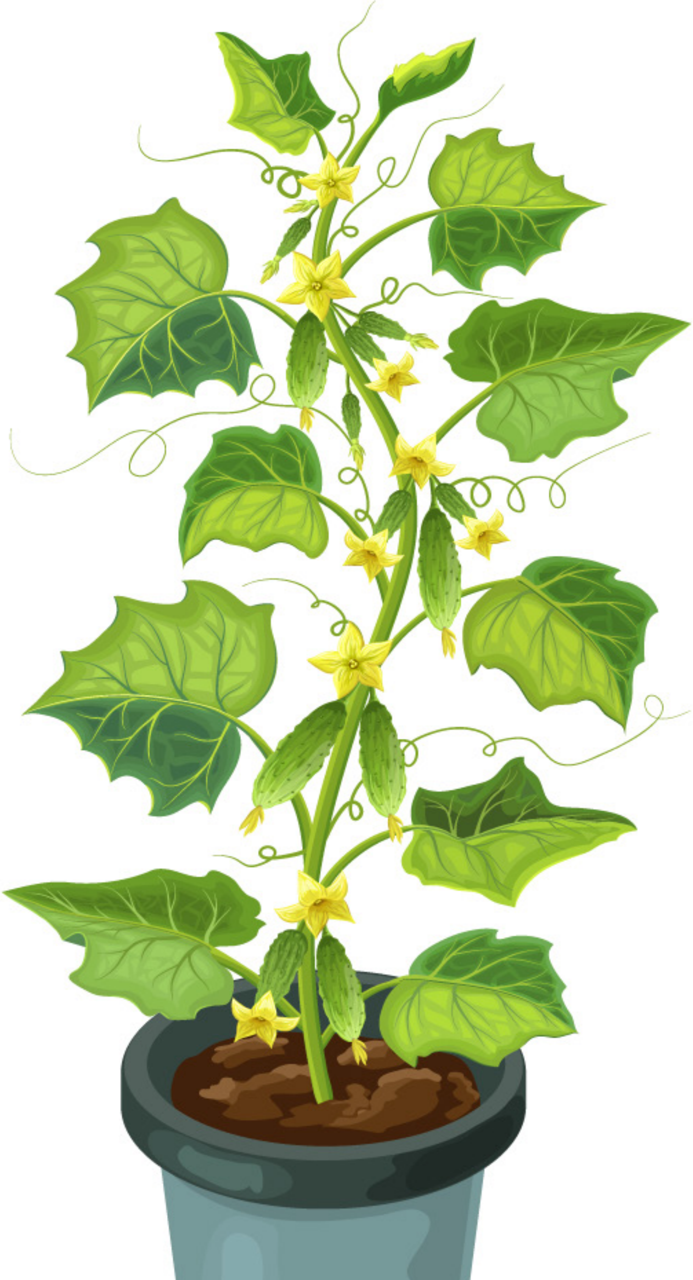
Still, already in the current form, the point clouds and the developed methods deliver quantitative information about the plant architecture. Furthermore, the point clouds can be segmented and this work could serve as an example on how to measure traits from the segmented point cloud. Future research could focus on data acquisition systems that reduce the level of occlusion, point cloud segmentation methods that improve the segmentation and can handle a higher level of phenotypic variation, and on methods to extract additional phenotypic measurements from the segmented point clouds.

5.5 Conclusion

The aim of this paper was to demonstrate the added value of 3D computer-vision based methods for measuring plant architecture. Previous work focused on the semantic segmentation of the 3D point clouds. The current work extends these methods by using the segmented point clouds to extract actual plant measurements. We have demonstrated that from the segmented point clouds, nodes can be detected and internode lengths can be estimated.

The first hypothesis that was tested, was that *"using 3D point clouds instead of 2D images allows to estimate internode lengths of curved plants with the same accuracy as obtained for straight plants"*. The results showed that based on our 3D method, there was indeed no difference in accuracy between internode lengths estimated for the curved and the straight plants. Moreover, we hypothesized that *"the error of the internode lengths estimated from the 3D point clouds is smaller than the error of the internode lengths estimated from the 2D images"*. Indeed, the results showed that the error obtained with the proposed 3D method in curved plants and in straight plants was significantly lower than the error obtained with the 2D method in straight plants.

A limitation of the 3D method was that not all nodes were detected due to occlusion and a lower resolution of the point clouds as compared to the 2D images. It was suggested to explore the possibilities of a combined approach, in which the advantages of both the 2D images and the 3D point clouds could be utilized. Still, in this paper, we have demonstrated that computer-vision based measurements of plant architecture in general, and more specifically of internode length, greatly benefit from the availability of 3D data.



Chapter 6

General discussion, reflections, and recommendations

General discussion, reflections, and recommendations

This chapter starts with a reflection on the main hypothesis of the thesis. An alternative view on this hypothesis is presented in section 6.2. In section 6.3, the measurement set-up and the cucumber data set are discussed, reflecting on the question whether a fair comparison between the developed 2D and 3D method could be made. Collecting proper training datasets by manually labelling the data is challenging, leading to an annotation bottleneck, on which section 6.4 reflects. The practical implications and societal relevance of the work is discussed in section 6.5. This chapter ends in section 6.6 with a reflection on the question that was raised in the title of this thesis: What is the point of plant phenotyping in three dimensions?

6.1 Reflection on the main hypothesis

The main objective of this thesis was to provide insight into the added value of 3D data and 3D-based methods for measuring plant-architectural traits. The internode length of cucumber plants was used as a model trait. A method to measure internode length based on 2D images was compared to a similar method based on 3D point clouds. The results of this comparison were used to evaluate the main hypothesis, which was:

> ***"phenotyping methods based on the acquisition and analysis of 3D data lead to improved phenotypic measurements of plant-architectural traits as compared to measurements obtained using 2D-based methods."***

In chapter 2, a method to measure the internode length based on 2D images was presented. The dataset was collected from multiple viewpoints around the plant. The node detection algorithm was able to detect most of the nodes in the images. By combining the results from the different viewpoints, the obtained F1-score of the 2D node detection algorithm was 0.93. However, the mean absolute error of the estimated internode lengths was 7.7 mm, which was less accurate than our reference measurements. Furthermore, these results were based on 9 out of the 12 plants. The remaining 3 plants had a curved growing pattern and as a result, these plants violated the assumption of a fixed plant-camera distance. Therefore, these plants were left out of the analysis. It was expected that a method based on 3D data would result in more accurate internode length estimates, as well as in the ability to accurately estimate internode lengths for the curved plants.

The 2D internode length method was based on an object detection algorithm. Although the same approach could have been followed by implementing a 3D object detection algorithm, a 3D semantic segmentation approach was followed instead. The first reason to do this was a practical one, being that 3D object detectors were, and are, less commonly available in comparison to 2D object detectors. Although this is also true for 3D semantic segmentation algorithms, the added value of a fully segmented point cloud as compared to only a detection of the nodes was considered to be high. However, implementation of these algorithms is not straightforward and 3D semantic segmentation in the plant domain is one of the novelties of this thesis.

The initial implementation of our method to segment the 3D point clouds, based on PointNet++, was presented in chapter 3. It was shown that the performance of the segmentation algorithm significantly increased for point clouds that were enriched with spectral data, as compared to point clouds having only geometric data. The best results were obtained for the plant parts for which many points were available. Classes for which only few points were available, like node, ovary, and tendril, showed a lower segmentation quality.

In chapter 4, an approach to tackle this class-imbalance problem was proposed. Due to memory limitations, point clouds were divided into smaller chunks, before being processed by the segmentation algorithm. In chapter 3, the division into chunks was designed in such a way that the entire point cloud was covered. By covering the entire point cloud, the class distribution of the training set was equal to the class distribution of the original point clouds. A novel method to divide the point clouds into chunks was developed, that allowed to focus on the minority classes. By doing so, a more balanced training set was obtained. It was shown that the segmentation of the minority classes drastically improved, also on the original, unbalanced, test data. At the same time, the segmentation performance for the majority classes remained equal, or was reduced only slightly.

Based on the improved segmentation performance, in chapter 5, the segmented point clouds were used to detect the location of the node objects. From the segmented point clouds, all points that were predicted to be of the class node were selected and clustered into node objects. The F1-score of the 3D node detection algorithm (0.87) was slightly lower than the F1-score of the 2D node detection algorithm (0.93). However, the estimated internode lengths were more accurate when using the 3D method as compared to the 2D method. Furthermore, while the 2D internode lengths for curved plants showed a higher error as compared to straight plants, the 3D internode lengths for straight and curved plants were equally accurate.

The results presented in this thesis clearly show that indeed, using 3D data leads to more accurate internode length estimates. More importantly, the use of 3D data allows to assess plant architecture for situations where a fixed plant-camera distance cannot be guaranteed. Even in our simplified experimental set-up, it turned oud that the assumption of a fixed plant-camera distance was not valid for all plants. In more complex environments, it will be even more important to be able to incorporate the distance between plant and camera in the algorithms. Furthermore, we only estimated internode length. For other traits related to the plant architecture, like leaf traits, there will be a higher level of variation in plant-camera distance. Based on these findings, the hypothesis of this thesis was confirmed, using 3D data leads to more accurate phenotypic measurements of plant architecture.

6.2 Another perspective on the hypothesis

The results presented above do support the main hypothesis. The 2D method was not suitable for all plants and had a larger error than the 3D method. However, in this section, the main hypothesis is revisited from another perspective. The main hypothesis was:

> ***"phenotyping methods based on the acquisition and analysis of 3D data lead to improved phenotypic measurements of plant-architectural traits as compared to measurements obtained using 2D-based methods."***

As such, the hypothesis searched for 'improved' measurements. A gap that was identified in hindsight, is that a definition of 'improved' was not given. The results presented above mainly focus on the accuracy of the measurements and on the ability to measure all plants. However, there are more aspects that play a role in digital phenotyping systems.

For example, the 3D method is clearly more complex than the 2D method. As an indicator, the 2D method was presented in 1 chapter, while presenting all components required for the 3D method took 3 chapters. Of course, the 3D method was based on semantic segmentation instead of on object detection, but we argue that a 2D semantic segmentation approach for these images should not be significantly more complex than the current approach using 2D object detection. Higher complexity in the data analysis inevitably comes at a higher cost. The same holds for the sensors required for the acquisition of 3D point clouds as compared to 2D images. In general, 3D-based systems are more complex and therefore more costly than 2D-based systems.

From an engineering perspective, it is not possible to assess whether the additional accuracy that can be achieved, justifies the additional complexity that is inherent to 3D-based systems. Therefore, from an engineering perspective, the hypothesis remains unanswered, as long as the requirements are unknown. To complement the work presented in this thesis, but also other research in the field, a discussion with the plant breeding community on requirements for digital phenotyping systems is highly recommended. Computer-vision based performance indicators in the light of practical applications, which are often reported, only tell part of the story. Additionally, for applied research, application-driven performance indicators in relation to requirements relevant for the application should be reported. Application-driven performance indicators could be, for example, the minimum number of internodes that should be measured, or the required accuracy of the estimated internode lengths in millimetres.

From a scientific perspective, we still argue that the work presented in this thesis is of value. Even if we do not yet know the specific requirements of an internode length estimation method, and therefore we cannot say which of the two presented methods is ‘better’, the comparison of the two methods provided valuable insights into the advantages and disadvantages of the two approaches. These insights contribute to future discussions about requirements of digital phenotyping systems.

6.3 The measurement set-up and the cucumber data set

The measurement set-up was equipped with an industrial 2D RGB camera for the 2D dataset and a PlantEye F500 DualScan system for the 3D dataset. The 2D and the 3D data were collected during one measurement campaign on the same plants. Also the manually collected reference measurements were the same for the 2D and the 3D method. This approach allowed a fair comparison of the accuracy of the estimated internode lengths.

However, the 2D images were taken from more viewpoints than the 3D point clouds, and the measuring principle of the 3D laser scanners led to a higher level of occlusion and missing data in the 3D dataset. Therefore, the comparability of the 2D and 3D method with respect to the number of visible and detected nodes is difficult. For the nodes that were visible in the data, the 2D method slightly outperformed the 3D method with respect to node detection. On the other hand, the 3D method outperformed the 2D method with respect to the accuracy of the measured internode lengths.

Due to the lower node visibility and detection in the 3D data, the ground truth data was used to determine if two nodes were part of the same internode. The use of ground truth data for this purpose limits practical implementation of the method. Therefore, especially for the 3D point clouds, reducing the level of occlusion is an important topic to explore. An interesting approach was presented in the work of Burusa et al. (2022), based on the principle of active vision. Instead of the predefined viewpoints that were used in this research, the viewpoints were selected per plant, based on an analysis of what viewpoint would reduce the level of occlusion the most. This was done either for whole-plant reconstruction, or with a focus on a specific plant parts.

In our experimental set-up, the distance between two plants in the row was 1 meter, to prevent that plants would occlude other plants. Although such a large distance between plants is a feasible scenario for specific phenotyping experiments, it is not feasible in commercial growing conditions. Especially for applications in highly cluttered environments, full plant reconstruction with tracking of objects over time is an interesting concept. Rincon et al. (2022) presented a method to detect tomato fruits. The detected fruits were registered in a world model, allowing tracking of the fruits over time. Similarly, it could be tested whether the nodes of a plant could be registered in such a model, to allow internode length estimation in cluttered environments. However, this is a complex task and it should be evaluated if growing plants in such cluttered environments is required for the phenotyping task at hand.

One of the interesting aspects of our dataset was that the plants could be followed to a plant height of approximately 2 meters, while many other works focused on smaller plants or seedlings (Nguyen et al., 2015; Shi et al., 2019; Yamamoto et al., 2016). However, the dataset that was used in this thesis consisted of only 12 plants of one cucumber variety. In that sense, the dataset did not include all variation in plant appearance that could be expected when implementing the developed methods at larger scale. It would be valuable to extend the dataset in future work to capture more variation in complexity. This would allow to test if our methods can indeed handle this level of variation. One of the challenges in these tests would be that the additional data would also have to be hand-labelled, which is still a challenging task. In the next section, we discuss this challenge as the ‘annotation bottleneck’.

6.4 The annotation bottleneck

The annotation bottleneck deals with the challenging task to obtain a hand-labelled dataset. The neural networks that were used in this thesis were supervised learning methods, meaning that a labelled data set was required to train the models. This training dataset was generated by manually drawing bounding boxes around the nodes in the case of the 2D method, and manually segmenting the entire point cloud in the case of the 3D methods. Especially for the 3D point clouds, this was a challenging and time-consuming task. Typically, each point cloud took around 10 minutes to label by hand.

Besides the time investment needed to generate hand-labelled data, the labelling process can be ambiguous, resulting in noisy or inaccurate training data. In chapter 3, the results of an experiment for which all point clouds were labelled twice were presented. The IoU between the two manually labelled datasets was used as an indication of the quality of the training data. Although for some classes there was a high correspondence between the two training sets, up to an IoU of 0.99 for the leaf, the IoU for the class node was as low as 0.49. This means that the node was labelled differently in both training sets.

In our experiment, the data was only labelled twice, by the same labeller. For 3D segmentation, but also for other tasks, it would be interesting to further analyse intra-observer variability and to add inter-observer variability. Having insight into the quality of the hand-labelled dataset opens the discussion on when the neural network is considered to be ‘right’. Possible strategies to count correct predictions include: right if the network agrees with most labellers, right if the network predicts any of the manually assigned labels, or, right if the network predicts all of the manually assigned labels, indicating that a point with many labels probably is a difficult or multi-interpretable instance.

The annotation bottleneck is also relevant for other computer vision tasks. A method to relieve this bottleneck based on active learning was presented in the work of Blok et al. (2022). In their work, first a neural network was trained on a small training dataset. The trained network was then used to predict the total dataset and select the images which would add the highest level of new information to the training dataset. Those images were then labelled and the network was retrained on the extended training set. This method allowed to obtain the same performance with less annotated images. Reducing the time needed for generating training data and increasing the quality of the training data are relevant fields of research that should be explored in future work.

While active learning can reduce the resources needed for collecting hand-labelled data, the learning process still depends on the availability of these labels and, especially for more complex tasks, on experts to label the data. Several approaches to reduce the annotation effort further are actively being researched. For example, semi-supervised learning was used for segmentation of remote sensing images, based on so-called scribble annotations. Instead of a pixel-wise labels of an entire image, only small parts of the images were labelled, drastically reducing labelling costs (Hua et al., 2022). One step further, Cao et al. (2019) presented an unsupervised approach for the segmentation of tomato plants in a greenhouse setting. Unsupervised learning uses no hand-labelled data for the training. Alternatively, self-supervised contrastive learning was successfully used on three public agricultural datasets (Güldenring & Nalpantidis, 2021). In the case of self-supervised learning, the labels are automatically generated and used to train a supervised network. Current work mainly shows the added value of unsupervised learning strategies to obtain pre-trained weights. The main advantage of this approach is that the amount of labelled data required for the final training of the network can be drastically reduced.

6.5 Practical implications and societal relevance

In chapter 1, an overview of how digital phenotyping technology could ease the phenotyping bottleneck was presented. The field of digital phenotyping deals with many different crops, sensors, and data analysis methods. However, based on the use case of this thesis, our methods were only tested on one variety of one crop, in one climate chamber, and based on one phenotyping system. As such, the dataset was rather limited. However, the only dataset-specific part in the developed methods was the training data. As long as a newly acquired dataset does not differ much from our dataset, the trained networks should still perform well, although we do recommend to always validate the performance on new datasets.

If the performance on a new dataset is below expectation, it is likely that the new dataset contains additional variation that was not covered in the original training set. The generalization of a model to other datasets depends on the ability to learn generic patterns instead of recognising individual training samples. In the work of Ruigrok et al. (2023), the generalisation error of a plant-detection model was evaluated. It was found that the generalisation error on new datasets could be reduced by fine-tuning an already trained model on a limited number of training samples from the new dataset. Especially when using our methods on datasets from other phenotyping systems in other environments, it is likely that the trained networks would benefit from retraining. When shifting to other crops, like tomato, it is expected that the principles of the node detection and internode length estimation methods will still work. However, the difference with the current training dataset will be larger and therefore, more training data will be required to retrain the model.

Besides applicability in other cucumber varieties, crops, or growing conditions, the methods presented in this thesis can be extended to measure other plant traits. The methods presented in chapter 3 and chapter 4 provide a segmented point cloud that can be used as input for the development of methods to assess these additional traits. One aspect that could increase the value of the segmented point cloud further is the identification of individual plant parts, known as instance detection. For the nodes, this step was added in chapter 5, but also for plant parts like the leaves and petioles this is a relevant addition. Having instance detection for these plant parts would allow not only to assess average leaf traits per plant, but also to asses those traits per individual leaf.

Of course, the work presented in this thesis has its limitations and being able to measure internode length or to segment a point cloud does not solve the big challenges of society in itself. Still, the development of computer-vision based methods to assess plants does contribute to solutions for these challenges, in particular in relation to food production. As a society, we should aim to provide good quality food for everyone, without damaging the environment we are living in. This means that we have to be able to adapt our plants to the continuously changing world. For example, salt and drought stress are two factors that potentially have a large negative effect on future crop productions. Furthermore, many tasks in agriculture depend on the availability of human labour, which is becoming more scarce and more expensive. Automated crop handling via robotics can partly replace human labour (Bac, 2015; Blok, 2022).

In successful plant breeding programs, improved plant varieties are developed that are able to deal with the changing environment in which they have to grow, or for example that are specifically designed for automated crop handling. In this thesis, we have explored and developed methods that can be used to generate high-quality phenotypic datasets for complex plant traits. Such datasets contribute to the success of breeding programs by increasing the genetic gain that can be achieved. The increased genetic gain, in the form of improved plant varieties that are developed within shorter amounts of time, contributes to a secure and sustainable food system.

6.6 General conclusion

In this thesis, we have presented and discussed automated methods for measuring internode length based on 2D and 3D data. The comparison of these methods provided valuable insights into the advantages and disadvantages of both methods. Furthermore, methods to obtain a completely segmented 3D point cloud, dealing with class imbalance, were developed. These methods were designed in a generic way and open up possibilities for developing more advanced phenotyping methods.

Digital plant phenotyping in three dimensions: What's the point?

The final paragraph reflects on the title of this thesis, raising the question what the point of digital plant phenotyping in three dimensions is? In one of the talks given in the past years, I summarized the answer to that question as 'use three dimensions if you must, but stick to two if you can'. The reason was that, although potentially more information could be extracted from 3D data, the extraction of the information will definitely be more complex. The answer to the question ultimately lies in the, not yet specified, required accuracy of the measurements. Still, based on the work presented in this thesis and the experience gained over the past years, I am convinced that the use case of plant architecture provides a clear example where using 3D data is a must.

References

Acquaah, G. (2007). Principles of Plant Genetics and Breeding: Second Edition. In *Principles of Plant Genetics and Breeding: Second Edition*. https://doi.org/10.1002/9781118313718

Alexey, A. B. (2018). *Darknet. Git code*. https://github.com/AlexeyAB/darknet/commit/cdd1cb0e8c4fda3671714bb5ad6ba1825cff16d1

Allard, R. W. (2019). *plant breeding*. Encyclopedia Brittanica. https://www.britannica.com/science/plant-breeding

Araus, J. L., Kefauver, S. C., Zaman-Allah, M., Olsen, M. S., & Cairns, J. E. (2018). Translating High-Throughput Phenotyping into Genetic Gain. *Trends in Plant Science*, *23*(5), 451–466. https://doi.org/10.1016/j.tplants.2018.02.001

Awada, L., Phillips, P. W. B., & Smyth, S. J. (2018). The adoption of automated phenotyping by plant breeders. *Euphytica*, *214*(8), 1–15. https://doi.org/10.1007/s10681-018-2226-z

Bac, C. W. (2015). *Improving obstacle awareness for robotic harvesting of sweet-pepper*. http://edepot.wur.nl/327202

Barth, R., IJsselmuiden, J., Hemming, J., & Henten, E. J. V. (2018). Data synthesis methods for semantic segmentation in agriculture: A Capsicum annuum dataset. *Computers and Electronics in Agriculture*, *144*(October 2017), 284–296. https://doi.org/10.1016/j.compag.2017.12.001

Blok, P. M. (2022). *Perception models for selective harvesting robots in fruit and vegetable production* [Wageningen University]. https://doi.org/10.18174/579739

Blok, P. M., Kootstra, G., Elghor, H. E., Diallo, B., van Evert, F. K., & van Henten, E. J. (2022). Active learning with MaskAL reduces annotation effort for training Mask R-CNN on a broccoli dataset with visually similar classes. *Computers and Electronics in Agriculture*, *197*(March), 106917. https://doi.org/10.1016/j.compag.2022.106917

Boogaard, F. P., Rongen, K. S. A. H., & Kootstra, G. W. (2020). Robust node detection and tracking in fruit-vegetable crops using deep learning and multi-view imaging. *Biosystems Engineering*, *192*, 117–132. https://doi.org/10.1016/j.biosystemseng.2020.01.023

Boogaard, F. P., van Henten, E. J., & Kootstra, G. (2021). Boosting plant-part segmentation of cucumber plants by enriching incomplete 3D point clouds with spectral data. *Biosystems Engineering*, *211*, 167–182. https://doi.org/10.1016/j.biosystemseng.2021.09.004

Boogaard, F. P., van Henten, E. J., & Kootstra, G. (2022). Improved Point-Cloud Segmentation for Plant Phenotyping Through Class-Dependent Sampling of Training Data to Battle Class Imbalance. *Frontiers in Plant Science*, *13*(March). https://doi.org/10.3389/fpls.2022.838190

Brugger, A., Behmann, J., Paulus, S., Luigs, H. G., Kuska, M. T., Schramowski, P., Kersting, K., Steiner, U., & Mahlein, A. K. (2019). Extending hyperspectral imaging for plant phenotyping to the UV-Range. *Remote Sensing*, *11*(12), 1–11. https://doi.org/10.3390/rs11121401

Burusa, A. K., van Henten, E. J., & Kootstra, G. (2022). *Attention-driven Active Vision for Efficient Reconstruction of Plants and Targeted Plant Parts*. http://arxiv.org/abs/2206.10274

Campello, R. J. G. B., Moulavi, D., & Sander, J. (2013). Density-based clustering based on hierarchical density estimates. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7819 LNAI*(PART 2), 160–172. https://doi.org/10.1007/978-3-642-37456-2_14

Cao, Q., & Xu, L. (2019). Unsupervised greenhouse tomato plant segmentation based on self-adaptive iterative latent Dirichlet allocation from surveillance camera. *Agronomy*, *9*(2). https://doi.org/10.3390/agronomy9020091

Chen, L., & Opara, U. L. (2013). Texture measurement approaches in fresh and processed foods - A review. *Food Research International*, *51*(2), 823–835. https://doi.org/10.1016/j.foodres.2013.01.046

CloudCompare. (2019). *CloudCompare (version 2.10.2)* (2.10.2). http://www.cloudcompare.org/

Cobb, J. N., Juma, R. U., Biswas, P. S., Arbelaez, J. D., Rutkoski, J., Atlin, G., Hagen, T., Quinn, M., & Ng, E. H. (2019). Enhancing the rate of genetic gain in public-sector plant breeding programs: lessons from the breeder's equation. *Theoretical and Applied Genetics : International Journal of Plant Breeding Research TA - TT -*, *132*(3), 627–645. https://doi.org/10.1007/s00122-019-03317-0 LK - https://wur.on.worldcat.org/oclc/8035714574

COCO. (2018). *COCO: Common Objects in Context - Detection Evaluation*. http://cocodataset.org/#detection-eval

Costa, C., Schurr, U., Loreto, F., Menesatti, P., & Carpentier, S. (2019). Plant phenotyping research trends, a science mapping approach. *Frontiers in Plant Science*, *9*(January), 1–11. https://doi.org/10.3389/fpls.2018.01933

Crain, J., Mondal, S., Rutkoski, J., Singh, R. P., & Poland, J. (2018). Combining High-Throughput Phenotyping and Genomic Information to Increase Prediction and Selection Accuracy in Wheat Breeding. *The Plant Genome*, *11*(1), 0. https://doi.org/10.3835/plantgenome2017.05.0043

Dhondt, S., Wuyts, N., & Inzé, D. (2013). Cell to whole-plant phenotyping: The best is yet to come. *Trends in Plant Science*, *18*(8), 1360–1385. https://doi.org/10.1016/j.tplants.2013.04.008

Dutagaci, H., Rasti, P., Galopin, G., & Rousseau, D. (2020). ROSE-X: An annotated data set for evaluation of 3D plant organ segmentation methods. *Plant Methods*, *16*(1), 1–14. https://doi.org/10.1186/s13007-020-00573-w

Dyrmann, M., Jørgensen, R. N., & Midtiby, 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

Fahlgren, N., Feldman, M., Gehan, M. A., Wilson, M. S., Shyu, C., Bryant, D. W., Hill, S. T., McEntee, C. J., Warnasooriya, S. N., Kumar, I., Ficor, T., Turnipseed, S., Gilbert, K. B., Brutnell, T. P., Carrington, J. C., Mockler, T. C., & Baxter, I. (2015). A versatile phenotyping system and analytics platform reveals diverse temporal responses to water availability in Setaria. *Molecular Plant*, *8*(10), 1520–1535. https://doi.org/10.1016/j.molp.2015.06.005

Fehr, W. R. (1991). Principles of Cultivar Development: Theory and Technique. *Agronomy Books*. https://doi.org/10.1097/00010694-198805000-00012

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., & Tsaftaris, S. A. (2018). Pheno-Deep Counter: a unified and versatile deep learning architecture for leaf counting. *Plant Journal*, *96*(4), 880–890. https://doi.org/10.1111/tpj.14064

Golbach, F., Kootstra, G., Damjanovic, S., Otten, G., & van de Zedde, R. (2016). Validation of plant part measurements using a 3D reconstruction method suitable for high-throughput seedling phenotyping. *Machine Vision and Applications*, *27*(5), 663–680. https://doi.org/10.1007/s00138-015-0727-5

Griffiths, D., & Boehm, J. (2019a). SynthCity: A large scale synthetic point cloud. *ArXiv*, 1–6.

Griffiths, D., & Boehm, J. (2019b). Weighted point cloud augmentation for neural network training data class-imbalance. *ArXiv*, *XLII*(June), 10–14.

Güldenring, R., & Nalpantidis, L. (2021). Self-supervised contrastive learning on agricultural images. *Computers and Electronics in Agriculture*, *191*(November), 106510. https://doi.org/10.1016/j.compag.2021.106510

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., & Bennamoun, M. (2020). *Deep Learning for 3D Point Clouds: A Survey*. *June*. https://doi.org/10.1109/tpami.2020.3005434

Heffner, E. L., Lorenz, A. J., Jannink, J. L., & Sorrells, M. E. (2010). Plant breeding with Genomic selection: Gain per unit time and cost. *Crop Science*, *50*(5), 1681–1690. https://doi.org/10.2135/cropsci2009.11.0662

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

Hua, Y., Marcos, D., Mou, L., Zhu, X. X., & Tuia, D. (2022). Semantic Segmentation of Remote Sensing Images With Sparse Annotations. *IEEE Geoscience and Remote Sensing Letters*, *19*(d), 1–5. https://doi.org/10.1109/LGRS.2021.3051053

Huang, S., Yan, H., Zhang, C., Wang, G., Acquah, S. J., Yu, J., Li, L., Ma, J., & Opoku Darko, R. (2020). Modeling evapotranspiration for cucumber plants based on the Shuttleworth-Wallace model in a Venlo-type greenhouse. *Agricultural Water Management*, *228*(March 2019), 105861. https://doi.org/10.1016/j.agwat.2019.105861

IMPERX. (2018). *IMPERX Industrial Cameras & Imaging Systems*. https://www.imperx.com/ccd-cameras/b4820/

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep Learning in Agriculture: A Survey. *Computers and Electronics in Agriculture*, *147*(1), 70–90. https://doi.org/10.1016/j.compag.2018.02.016

Kjaer, K., & Ottosen, C.-O. (2015). 3D Laser Triangulation for Plant Phenotyping in Challenging Environments. *Sensors*, *15*(6), 13533–13547. https://doi.org/10.3390/s150613533

LeCun, Y., Bengio, Y., Hinton, G., Y., L., Y., B., & G., H. (2015). Deep learning Review. *Nature*, *521*. https://doi.org/10.1038/nature14539

Li, L., Zhang, Q., & Huang, D. (2014). A review of imaging techniques for plant phenotyping. *Sensors (Switzerland)*, *14*(11), 20078–20111. https://doi.org/10.3390/s141120078

Li, Z., Guo, R., Li, M., Chen, Y., & Li, G. (2020). A review of computer vision technologies for plant phenotyping. *Computers and Electronics in Agriculture*, *176*(July), 105672. https://doi.org/10.1016/j.compag.2020.105672

Lin, H. I., & Nguyen, M. C. (2020). Boosting minority class prediction on imbalanced point cloud data. *Applied Sciences (Switzerland)*, *10*(3). https://doi.org/10.3390/app10030973

Lin, T., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). *Focal Loss for Dense Object Detection*. http://arxiv.org/abs/1708.02002

Litvin, A. G. (2009). *Interaction of Drought Stress and Gibberellin Metabolism on stem elongation in tomato*. University of Georgia.

Litvin, A. G., Van Iersel, M. W., & Malladi, A. (2016). Drought stress reduces stem elongation and alters gibberellin-related gene expression during vegetative growth of tomato. *Journal of the American Society for Horticultural Science*, *141*(6), 591–597. https://doi.org/10.21273/JASHS03913-16

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 in Plant Science*, *8*(December), 1–3. https://doi.org/10.3389/fpls.2017.02181

Mele, G., & Gargiulo, L. (2020). Automatic cell identification and counting of leaf epidermis for plant phenotyping. *MethodsX*, *7*. https://doi.org/10.1016/j.mex.2020.100860

Milioto, A., Vizzo, I., Behley, J., & Stachniss, C. (2019). RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation. *IEEE International Conference on Intelligent Robots and Systems*, *i*, 4213–4220. https://doi.org/10.1109/IROS40897.2019.8967762

Minervini, M., Scharr, H., & Tsaftaris, 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. 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*, *July*.

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

Nicolaï, B. M., Defraeye, T., De Ketelaere, B., Herremans, E., Hertog, M. L. a T. M., Saeys, W., Torricelli, A., Vandendriessche, T., & Verboven, P. (2014). Nondestructive measurement of fruit and vegetable quality. *Annual Review of Food Science and Technology*, *5*(1), 285–312. https://doi.org/10.1146/annurev-food-030713-092410

NVIDIA Corporation. (2018a). *CUDA Toolkit*. https://developer.nvidia.com/cuda-toolkit

NVIDIA Corporation. (2018b). *GEFORCE GTX 1080 Ti*. https://www.nvidia.nl/graphics-cards/geforce/pascal/gtx-1080-ti

NVIDIA Corporation. (2018c). *NVIDIA cuDNN*. https://developer.nvidia.com/cudnn

OpenCV team. (2018). *OpenCV (Open Source Computer Vision Library)*. https://opencv.org/

Paulus, S. (2019). Measuring crops in 3D: using geometry for plant phenotyping. *Plant Methods TA - TT -*, *15*, 103. https://doi.org/10.1186/s13007-019-0490-0 LK - https://wur.on.worldcat.org/oclc/8227048808

Perreault, A. (2018). *How England Got Its Curvy Cucumbers Straightened Out - Gastro Obscura*. https://www.atlasobscura.com/articles/historic-gardening-tools

Phenospex. (2017). *PlantEye F500*. https://phenospex.com/products/plant-phenotyping/planteye-f500-multispectral-3d-laser-scanner/

Poliyapram, V., Wang, W., & Nakamura, R. (2019). A point-wise LiDAR and image multimodal fusion network (PMNet) for aerial point cloud 3D semantic segmentation. *Remote Sensing*, *11*(24). https://doi.org/10.3390/rs11242961

Pound, M. P., Atkinson, J. A., Townsend, A. J., Wilson, M. H., Griffiths, M., Jackson, A. S., Bulat, A., Tzimiropoulos, G., Wells, D. M., Murchie, E. H., Pridmore, T. P., & French, A. P. (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. *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, *2018-Janua*, 2055–2063. https://doi.org/10.1109/ICCVW.2017.241

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2016). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *2016 Fourth International Conference on 3D Vision (3DV)*, 601–610. https://doi.org/10.1109/3DV.2016.68

Qi, C. R., Yi, L. (Eric), Su, H., & Guibas, L. J. (2018). *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*. https://github.com/charlesq34/pointnet2

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*. http://arxiv.org/abs/1706.02413

R Core Team. (2018). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing.

Rebetzke, G. J., Jimenez-Berni, J., Fischer, R. A., Deery, D. M., & Smith, D. J. (2019). Review: High-throughput phenotyping to enhance the use of crop genetic resources. *Plant Science*, *282*(April 2018), 40–48. https://doi.org/10.1016/j.plantsci.2018.06.017

Redmon, J. (2016). *Darknet: Open Source Neural Networks in C*. https://pjreddie.com/darknet/

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 6517–6525. https://doi.org/10.1109/CVPR.2017.690

Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. https://doi.org/10.1109/CVPR.2017.690

Reinhardt, D., & Kuhlemeier, C. (2002). Plant architecture. *EMBO Reports*, *3*(9), 846–851. https://doi.org/10.1093/embo-reports/kvf177

Rincon, D. R., van Henten, E. J., & Kootstra, G. (2022). *Development and evaluation of automated localization and reconstruction of all fruits on tomato plants in a greenhouse based on multi-view perception and 3D multi-object tracking*. http://arxiv.org/abs/2211.02760

Roitsch, T., Cabrera-Bosquet, L., Fournier, A., Ghamkhar, K., Jiménez-Berni, J., Pinto, F., & Ober, E. S. (2019). Review: New sensors and data-driven approaches—A path to next generation phenomics. *Plant Science*, *282*(January), 2–10. https://doi.org/10.1016/j.plantsci.2019.01.011

Rosenberg, A., & Hirschberg, J. (2007). V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. *Proceedings of the 2007 Joint Conference on Emprical Methods in Natural Language Processing and Computational Natural Language Learning*, *June*, 410–420. https://doi.org/10.7916/D80V8N84

Rossi, R., Costafreda-Aumedes, S., Summerer, S., Moriondo, M., Leolini, L., Cellini, F., Bindi, M., & Petrozza, A. (2022). A comparison of high-throughput imaging methods for quantifying plant growth traits and estimating above-ground biomass accumulation. *European Journal of Agronomy*, *141*(May), 126634. https://doi.org/10.1016/j.eja.2022.126634

Ruigrok, T., van Henten, E. J., & Kootstra, G. (2023). Improved generalization of a plant-detection model for precision weed control. *Computers and Electronics in Agriculture*, *204*(November 2022), 107554. https://doi.org/10.1016/j.compag.2022.107554

Rusu, R. B., & Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). *IEEE International Conference on Robotics and Automation (ICRA)*. https://doi.org/10.1073/pnas.74.3.1167

Sander, R. (2020). *Sparse Data Fusion and Class Imbalance Correction Techniques for Efficient Multi-Class Point Cloud Semantic Segmentation Sparse Data Fusion and Class Imbalance Correction Techniques for Efficient Multi-Class Point Cloud Semantic Segmentation*. *February*, 0–6. https://doi.org/10.13140/RG.2.2.12077.03042

Sasaki, Y. (2007). *The truth of the F-measure*. 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

Shi, W., van de Zedde, R., Jiang, H., & Kootstra, G. (2019). Plant-part segmentation using deep learning and multi-view vision. *Biosystems Engineering*, *187*, 81–95. https://doi.org/10.1016/j.biosystemseng.2019.08.014

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. http://www.scienceandnature.org/GJBB_Vol2(4)2013/GJBB-V2(4)2013-1.pdf

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. https://doi.org/10.1016/j.tplants.2018.07.004

Smith, D. T., Potgieter, A. B., & Chapman, S. C. (2021). Scaling up high-throughput phenotyping for abiotic stress selection in the field. *Theoretical and Applied Genetics*, *134*(6), 1845–1866. https://doi.org/10.1007/s00122-021-03864-5

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

Suter, L., & Widmer, A. (2013). Phenotypic effects of salt and heat stress over three generations in Arabidopsis thaliana. *PLoS ONE*, *8*(11), 1–13. https://doi.org/10.1371/journal.pone.0080819

Tanner, F., Tonn, S., de Wit, J., Van den Ackerveken, G., Berger, B., & Plett, D. (2022). Sensor-based phenotyping of above-ground plant-pathogen interactions. *Plant Methods*, *18*(1), 1–18. https://doi.org/10.1186/s13007-022-00853-7

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

Tripodi, P., Nicastro, N., & Pane, C. (2022). Digital applications and artificial intelligence in agriculture toward next-generation plant phenotyping. *Crop and Pasture Science*. https://doi.org/10.1071/CP21387

Turgut, K., Dutagaci, H., Galopin, G., & Rousseau, D. (2020). *Segmentation of structural parts of rosebush plants with 3D point-based deep learning methods*. http://arxiv.org/abs/2012.11489

Turgut, K., Dutagaci, H., Galopin, G., & Rousseau, D. (2022). Segmentation of structural parts of rosebush plants with 3D point-based deep learning methods. *Plant Methods*, *18*(1), 20. https://doi.org/10.1186/s13007-022-00857-3

Tzutalin. (2015). *LabelImg. Git code*. https://github.com/tzutalin/labelImg

Ubbens, J. R., & Stavness, I. (2017). Deep Plant Phenomics: A Deep Learning Platform for Complex Plant Phenotyping Tasks. *Frontiers in Plant Science*, *8*(July). https://doi.org/10.3389/fpls.2017.01190

van der Heijden, G., Song, Y., Horgan, G., Polder, G., Dieleman, A., Bink, M., Palloix, A., van Eeuwijk, F., & Glasbey, C. (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

Vázquez-Arellano, M., Griepentrog, H. W., Reiser, D., & Paraforos, D. S. (2016). 3-D imaging systems for agricultural applications—a review. *Sensors (Switzerland)*, *16*(5). https://doi.org/10.3390/s16050618

Ward, D., Moghadam, P., & Hudson, N. (2019). Deep leaf segmentation using synthetic data. *British Machine Vision Conference 2018, BMVC 2018*.

Wieczorek, A. M., & Wright, M. G. (2012). History of Agricultural Biotechnology : How Crop Development has Evolved. *Nature Education Knowledge*, *3(10):9*.

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

WIWAM. (2022). *WIWAM Phenotyping robots*. https://www.wiwam.be/projects/

Wu, C., Zeng, R., Pan, J., Wang, C. C. L., & Liu, Y. J. (2019). Plant phenotyping by deep-learning-based planner for multi-robots. *IEEE Robotics and Automation Letters*, *4*(4), 3113–3120. https://doi.org/10.1109/LRA.2019.2924125

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

Yang, W., Feng, H., Zhang, X., Zhang, J., Doonan, J. H., Batchelor, W. D., Xiong, L., & Yan, J. (2020). Crop Phenomics and High-Throughput Phenotyping: Past Decades, Current Challenges, and Future Perspectives. *Molecular Plant*, *13*(2), 187–214. https://doi.org/10.1016/j.molp.2020.01.008

Yao, L., Van De Zedde, R., & Kowalchuk, G. (2021). Recent developments and potential of robotics in plant eco-phenotyping. *Emerging Topics in Life Sciences*, *5*(2), 289–300. https://doi.org/10.1042/ETLS20200275

Yi, F., & Moon, I. (2014). K-means based Clustering Method with a Fixed Number of Cluster Members. *Journal of Korea Multimedia Society*, *17*(10), 1160–1170. https://doi.org/10.9717/kmms.2014.17.10.1160

Yol, E., Toker, C., & Uzun, B. (2015). Phenomics in crop plants: Trends, options and limitations. In *Phenomics in Crop Plants: Trends, Options and Limitations*. https://doi.org/10.1007/978-81-322-2226-2

Zhu, X., Leiser, W. L., Hahn, V., & Würschum, T. (2021). Phenomic selection is competitive with genomic selection for breeding of complex traits. *Plant Phenome Journal*, *4*(1), 1–21. https://doi.org/10.1002/ppj2.20027

Summary

Plant breeding is the art and science of developing improved plant varieties and is essential for global food security. In plant breeding programs, new varieties can be developed that are more resistant to diseases, more tolerant to stress factors such as drought and salinity, or that produce a higher yield with the same or a smaller amount of resources required as input.

At the heart of a plant breeding program is the process to find the optimal combination of plants to cross. The success of this process can be quantified as the genetic gain of the breeding program. In **chapter 1**, the phenotyping bottleneck is identified as a factor that limits the genetic gain. The phenotyping bottleneck is the lack of high-quality phenotypic datasets of large plant populations. The concept of digital phenotyping is then presented as a technological approach that is able to ease the phenotyping bottleneck.

The essence of digital plant phenotyping is to use sensors that capture data of plants and to analyse this data to extract relevant phenotypic measurements. The potential benefits of implementing digital phenotyping technologies can be summarised in three aspects. First, automation of digital phenotyping systems drastically reduces the required labour per data point, meaning that it becomes feasible to deal with larger plant populations. Second, the accuracy and the objectivity of the measurements increases as compared to human observations. Third, traits that are not visible to the human eye can be assessed by using techniques such as hyperspectral imaging, MRI, or X-ray. Digital phenotyping can contribute to an increased genetic gain and as such to the development of improved plant varieties that become available to growers within shorter amounts of time.

This thesis explores digital phenotyping technology to measure plant-architectural traits. Cucumber was used as a model crop and internode length was used as a model trait. Furthermore, 3D sensor data and the corresponding deep-learning based methods to translate the 3D data into plant measurements were selected as promising technologies. The developed 3D-based methods were compared to a 2D-based approach, to gain insight into the added value of 3D data for digital plant phenotyping.

First, a method to estimate internode length based on 2D images was developed. This method is presented in **chapter 2**. The first step was to detect the nodes in the individual images based on a deep convolutional neural network. Multiple viewpoints around the plant were used to reduce the number of occluded nodes.

The different viewpoints were combined into one reference coordinate system, after which affinity propagation was used to cluster multiple detections of the same node, resulting in a set of detected nodes. The 2D Euclidean distance between two consecutive nodes was then used as an estimate of the internode length.

The results presented in chapter 2 clearly indicated the added value of having multiple viewpoints for the number of nodes that was visible in the dataset. The trained neural network was able to reliably detect these nodes. The estimated internode lengths were compared to a fast reference measurement and to an accurate reference measurement. The fast reference measurement was based on dividing the plant length by the number of internodes to obtain an average internode length, similar to current practice. The accurate reference measurement was based on measuring all internodes with a measuring tape. The results of our 2D method were more accurate than the fast reference measurement, but less accurate than the accurate reference measurement.

Besides the accuracy of the estimated internode lengths, the error of the 2D method was significantly larger for plants with a curved growing pattern. This was the case for 25% of the plants, which were therefore left out of the analysis. It was hypothesized that a method based on 3D data would be able to increase the accuracy of the estimated internode lengths, and would be able to provide the same level of accuracy for straight as well as for curved plants. Therefore, the remaining chapters focus on methods to deal with 3D point clouds to measure plant architecture.

In **chapter 3**, the focus was on semantic segmentation of the 3D point clouds, meaning that for each point in the point cloud it was predicted to which plant organ it belonged. The prediction was based on a neural network architecture called PointNet++. Due to memory limitations, the point clouds could not be processed at once. Therefore, the point clouds were divided into smaller chunks that were segmented by the network. The segmented chunks were then recombined to obtain a segmented point cloud of the entire plant. To obtain a training dataset for the network and to be able to evaluate the predicted segmentation, all point clouds were manually segmented first. One of the experiments presented in this chapter was based on manually segmenting all point clouds a second time. This allowed to quantify the intra-observer variability between the first and the second time the data was hand-labelled. The low agreement for some classes clearly showed that consistent annotation of point clouds is a challenging task.

The results of this chapter demonstrated the ability of a PointNet++ -based method to segment point clouds of cucumber plants. The network was trained on the point clouds with only geometric data and on the point clouds enriched with spectral data. It was shown that the availability of spectral data significantly improved the segmentation of stem, petiole, leaf, tendril and non-plant objects, as well as the overall segmentation quality. Combining the two labelled datasets resulted in a small but significant improvement of the segmentation quality for the stem, petiole, node and ovary. Finally, we showed that a careful design of the plant phenotyping experiment improved the segmentation quality. For all classes, the segmentation quality increased when the segmentation task contained fewer classes.

One of the findings that was reported in chapter 3, was that the segmentation task suffered from a class-imbalance problem. Therefore, in **chapter 4**, a method to improve the recognition of the smaller classes is presented. Inspired by methods to handle class imbalance in 2D images, a method to focus the attention of the 3D neural network on the smaller classes was proposed. In this method, we made use of the division of the point clouds into smaller chunks to create a training dataset in which the different classes were more equally represented, while maintaining the original structures that were present in the data. The trained network was then tested on the original data.

As hypothesized, the results of this chapter demonstrated that class-dependent sampling of the training data improved the class balance in the training set. The segmentation performance on the original, imbalanced, test data was significantly improved. As expected, the biggest improvements were found for the smallest classes. The percentage of correctly predicted ‘node’ points increased by 46.0 percentage points.

The improved segmentation of the class ‘node’ from chapter 4 was used in **chapter 5** to test if the improved segmentation was sufficient to detect the node-objects in the point clouds. The points that the segmentation algorithm predicted as node were first selected from the point cloud. A clustering algorithm was used to group points that belonged to the same node, moving from segmented points to instance detection of plant parts. Based on the detected nodes, the 3D internode length was estimated and compared to the 2D internode length estimates obtained in chapter 2.

The results showed that the error obtained with the proposed 3D method in curved plants and in straight plants was significantly lower than the error obtained with the 2D method in straight plants. Furthermore, the error of the 3D method for curved plants was not significantly different from the error obtained for straight plants. This observation confirmed that using 3D point clouds instead of 2D images allows to estimate internode lengths of curved plants with the same accuracy as obtained for straight plants.

A limitation of the 3D method was that not all nodes were detected due to occlusion and a lower resolution of the point clouds as compared to the 2D images. It was suggested to explore the possibilities of a combined approach, in which the advantages of both the 2D images and the 3D point clouds could be utilized. Still, we demonstrated that computer-vision based measurements of plant architecture in general, and more specifically of internode length, greatly benefit from the availability of 3D data.

Finally, in **chapter 6**, the conclusions of the thesis are summarized and a general discussion of the research is presented. The results that were presented in the previous chapters confirmed the main hypothesis of the thesis: *"phenotyping methods based on the acquisition and analysis of 3D data lead to improved phenotypic measurements of plant-architectural traits as compared to measurements obtained using 2D-based methods"*. However, it was also noted that using 3D data inevitably leads to more complex methods. Whether the increased accuracy justifies the additional complexity of the method was discussed in this chapter. The thesis is concluded with a reflection on the question that was raised in the title of this thesis: "Digital plant phenotyping in three dimensions – what's the point?".

Acknowledgements

Finally, after a long period of hard work, my thesis is finished and has been approved by the thesis committee. This means that the only step left to finish my PhD is the defence. I am confidently looking forward to this last step, although the nerves will probably kick in somewhere in the coming weeks. In this final section, I would like to look back on the journey that I have been on and to express my gratitude to a number of people who played an important role in the successful completion of this journey.

In 2005, I decided to move to Wageningen and started the bachelor programme on agricultural engineering, agrotechnologie. I enjoyed my time in Wageningen, including several side activities, and after the bachelor I continued with the master programme. From this programme, I specifically remember a number of lectures given for the course ‘Automation for bio-production’. This course was, in my opinion, one of the best courses I attended and these lectures really boosted my enthusiasm for our field. Eventually, my interest in automation for agriculture and horticulture led to an MSc thesis with as topic the control of a robotic arm for sweet pepper harvesting.

Around the time of my graduation, the first talks about doing a PhD started. However, I was convinced that I wanted to spend some time outside of academia first and I joined Rijk Zwaan. After a few years, at the PhD defence of Wouter Bac in 2015, the talks about a PhD project continued. Several meetings took place over the years in which we exchanged plans and I had a hard time making up my mind. But, finally, the decision was made to ‘do a PhD’ and in 2018 the journey started.

Enthusiastically, I started working on the proposal and the first paper. Most of the time, the enthusiasm remained, although I also must admit that at times I have regretted starting a PhD quite a bit. Especially at times when I couldn’t motivate myself to keep working on those papers, or if the review process took forever. Luckily, there were many exciting life-events that I could enjoy in the same period as I was doing my PhD research, such as marrying Martiene, buying and renovating a house, and becoming father of Tim and Merel. Around the time Merel was born, the finish line of my PhD slowly came into sight and with each step I got more confident that I could make it. It took one final sprint of about a year to complete my PhD research in a way that I am proud of.

Now it is time to spend some words to thank the people who helped me to accomplish this. I would like to start with the person who gave these inspiring lectures, which happens to be the same person who was the supervisor of my MSc thesis and with whom I have had all these discussions about whether or not I should start a PhD project. It is also the same person who is my promotor today. Eldert, as you can see, you have played quite a big role. You motivated me, you were critical about the things we were doing which sharpened my mind, you showed me the bigger picture of our work, and when needed, you were pragmatic to make sure that things kept moving forward. I am happy and grateful that you wanted to be my promotor. Thank you!

Gert, my co-promotor, I also owe you a big thanks! You have been involved from the beginning of the PhD project and I really enjoyed your enthusiasm and optimism. You were always able to see the positive side of things. When I saw mountains, you made them appear like small hills. I remember when we thought of new ideas to add in a paper, your estimation would typically be that it wouldn’t take more than a few days of work. My estimation would then be that it would take a few weeks. Most of the times we met in the middle :) Gert, thanks a lot for the great times we have had together!

There is no PhD defence without a thesis committee. To all members of this committee, thank you for taking the time to read my thesis and for coming to Wageningen for the defence. I am looking forward to our discussion and reflections. Thanks to my paranymphs Mark and Thijs for the support during the defence.

I also would like to express my gratitude to Rijk Zwaan. From the beginning, it was clear to me that I wanted to do this PhD as part of the Rijk Zwaan family. Tonny, you were the first one at Rijk Zwaan to discuss my PhD research with. You have always supported me and have been following my work even after changing your position in Rijk Zwaan. You helped me in finding the right ways to get this project approved. Thanks for that! Björn, you were also part of the Rijk Zwaan supervision team. Thanks for your constructive feedback on papers and on helping me to take the breeding perspective into account. Of course, I would also like to thank the board of directors and Jack. Without your support, this PhD project would not have been possible. Mike, Jan Willem, and Erwin, as current IMS R&D team leads, thanks for giving me the time to finish this project.

Of course, there were many more colleagues who were involved, some more content-related, and others just to have a nice chat. My department and team IMS, IMS R&D and AIA, thanks for listening to my sometimes maybe boring updates ('the last four weeks I have been working on a paper'). I am looking forward to discuss other projects with you again. Arjan, we had one talk about one and a half year ago when I was close to quitting and you convinced me to keep going. Thanks for that! Ed, thanks for always stopping by the office to check if the thesis was already finished. It goes way too far to mention all colleagues personally, but to all who supported me in one way or the other, thanks!

Kees, we have spent many hours in the climate chamber in Stampersgat. Thanks for all the work that you have done there, but also for all the nice talks that we have had. You were of great support for my PhD project!

At the beginning of my PhD, in the pre-covid times, I drove to Wageningen every Thursday to work at the Farm Technology group. It was really nice to be in the PhD room every week, talking to people who were in the same boat. Although the distance between Sint-Annaland and Wageningen was too big to join all activities of the group, I have enjoyed the drinks and barbecues where I could be present a lot. Thanks to all FTE-colleagues for these nice moments!

As Pieter already mentioned in the acknowledgements of his PhD thesis, the week in Amsterdam together with Thijs was surely the highlight of our TSP. Pieter, dr. Blok, thanks a lot for sharing our struggles and our achievements. Thijs, you are by far the PhD candidate with whom I have had the most conversations. Sometimes about work, but mostly about all kinds of other things. Looking forward to a reunion in Amsterdam!

Music has always been an important part of my life. During my PhD, I have played in several British-style brass bands and fanfare orchestras. When I joined the last one, several people asked how I could combine another weekly rehearsal with our family life and of course, the PhD. But to me, these rehearsals were the rare moments where I wouldn't think about my PhD at all, making them a great way to clear my mind. Excelsior, Euterpe, Bacchus, and Accelerando, thanks for all the great musical moments and for helping me to forget my point clouds every now and then.

Pa en ma, ook jullie wil ik bedanken voor jullie steun. Niet alleen in de jaren dat ik aan mijn PhD heb gewerkt, maar ook in alle jaren daarvoor. Ik ben heel dankbaar voor alle kansen die ik dankzij jullie gekregen heb en ik vind het fijn dat jullie erbij kunnen zijn. Natuurlijk ook dank aan de rest van het gezin. Willem, Bram en Femke, Betsy en Leen, we gaan zeker nog een keutje doen om de afronding van mijn PhD te vieren.

En natuurlijk: Merel, Tim en Martiene. Merel, je zult er wellicht het jouwe van gedacht hebben, al kon je er natuurlijk nog niet veel over zeggen. Tim, voor jou ging dat al een stukje beter. Als ik weer eens weg moest voor een te lange dag dan vroeg ik je: “Tim, weet je wat papa gaat doen?”, waarop je antwoordde: “Boek schrijven. En groente maken.” Martiene, jij was de stille kracht, die ons prachtige gezin draaiend hield. Het is een cliché, maar zonder jou was het niet gelukt. Ik kijk er van harte naar uit om heel veel meer avonturen met jullie te gaan beleven!

PE&RC Training and education statement

With the training and education activities listed below the PhD candidate has complied with the requirements set by the C.T. de Wit Graduate School for Production Ecology and Resource Conservation (PE&RC) which comprises of a minimum total of 32 ECTS (= 22 weeks of activities)

Literature review / project proposal (6 ECTS)

- Project proposal: plant architecture measurements from incomplete 3-dimensional multispectral sensor data in fruit crops
- Literature review: class imbalance in point-based neural networks

Post-graduate courses (10 ECTS)

- Phenotyping technologies in plant-environment interactions – integrated analysis of omics data
  NOVA University Network, SLU Alnarp, Sweden (2018)
- Deep learning specialization
  Coursera (2018)
- Computer vision by learning
  ASCI: Advanced School for Computing and Imaging, Amsterdam (2019)

Invited review of journal manuscripts (5 ECTS)

- Computers and Electronics in Agriculture: automated image annotation (2020)
- Biosystems Engineering: high-throughput plant phenotyping (2020)
- Computers and Electronics in Agriculture: MRI for plant phenotyping (2021)
- Computers and Electronics in Agriculture: non-invasive plant phenotyping under field conditions (2021)
- Computers and Electronics in Agriculture: high-throughput point cloud segmentation for plant phenotyping (2021)

Competence strengthening / skills courses (2.45 ECTS)

- Brain friendly working & writing; *Wageningen Graduate Schools (2019)*
- Efficient writing strategies; *Wageningen in'to Languages (2019)*
- Supervising BSc & MSc thesis students; *Education Support Centre (2019)*
- How to present online; *The floor is yours (2020)*
- LateX workshop; *PE&RC (2020)*

Scientific integrity/ethics in science activities (0.6 ECTS)

- Scientific integrity; *Wageningen Graduate Schools (2020)*

PE&RC Annual meetings, seminars and the PE&RC weekend (1.2 ECTS)

- PE&RC Day (2018)
- PE&RC Workshop carousel (2019)
- PE&RC Last years event (2020)
- PE&RC Afternoon (2020)
- PE&RC Workshop Carousel (2021)

Discussion groups / local seminars or scientific meetings (7.9 ECTS)

- Rijk Zwaan R&D symposium; oral presentation (2018)
- Deep learning topic group (2019)
- Rijk Zwaan breeding symposium; oral presentation (2019)
- The Dutch conference on computer vision; poster presentation (2019)
- Rijk Zwaan R&D symposium; oral presentation (2022)

International symposia, workshops and conferences (5.2 ECTS)

- Plant breeding and biotechnology symposium;
  Oral presentation; SLU and WUR, Wageningen (2019)
- European Conference on Agricultural Engineering – New challenges for agricultural engineering towards a digital world;
  Oral presentation; Évora, Portugal (2021)

Societally relevant exposure (0.3 ECTS)

- Lecture on digital plant phenotyping for minor smart farming of HAS Hogeschool (2018)

Lecturing/supervision of practicals/tutorials (1.2 ECTS)

- Sensor technology (2019)
- Summer school image analysis for plant phenotyping (2019)

BSc/MSc thesis supervision (3 ECTS)

- Estimating internode length in cucumber plants based on 2D images (2018)

<u>Total credits: 42.85 ECTS</u>

Frans Boogaard gratefully acknowledges Rijk Zwaan for the funding of his position as PhD-candidate. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Financial support from Wageningen University and Rijk Zwaan for printing this thesis is gratefully acknowledged.

Cover design by:	Ron Zijlmans | RON Graphic Power | www.ron.nu
Printed by:	ProefschriftMaken.nl