Available online at www.sciencedirect.com

ScienceDirect

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

Research Paper

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

Frans P. Boogaard [a,b,*], Eldert J. van Henten [a], Gert Kootstra [a]

[a] Wageningen University & Research, Farm Technology Group, P.O. Box 16, 6700 AA, Wageningen, the Netherlands
[b] Rijk Zwaan Breeding, Eerste Kruisweg 9, 4793 RS, Fijnaart, the Netherlands

ARTICLE INFO

Article history:
Received 31 May 2021
Received in revised form
3 September 2021
Accepted 10 September 2021

Keywords:
Plant-part segmentation
Point cloud
Spectral data
Phenotyping
Deep learning

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 (DL). 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.

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

* Corresponding author. Wageningen University & Research, Farm Technology Group, P.O. Box 16, 6700 AA, Wageningen, the Netherlands.
E-mail address: frans.boogaard@wur.nl (F.P. Boogaard).
https://doi.org/10.1016/j.biosystemseng.2021.09.004

Nomenclature

2D/3D	2-/3-dimensional
DL	Deep learning
IoU	Intersection over Union
RGB	Spectral features (red, green and blue reflectance)
TP, FP, TN, FN	True Positive, False Positive, True Negative, False Negative
XYZ	Geometric features (x, y and z)
XYZRGB	The combination of geometric and spectral features

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 (DL)-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, DL-based methods outperform feature-based methods. Computer-vision techniques based on DL 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 focussing on plant materials. In another recent comparison of point-based segmentation using DL (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.

1.1. *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 DL-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, 2019).

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.

2. Materials and 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 2.1. The segmentation method is introduced in section 2.2, including a description of how we processed our data. In section 2.3, the evaluation criteria used to measure the performance of our method are introduced. Finally, in section 2.4, the experiments performed to answer the research questions are explained in more detail.

2.1. Data

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 m. 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 7–9 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 Fig. 1.

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 Fig. 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 min. 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 plants $\times$ 2 sides $\times$ 11 days $=$ 264 point clouds. A schematic overview of the layout of the climate chamber, including the scan positions and plant IDs, is shown in Fig. 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 Fig. 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 Fig. 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

Fig. 1 – Examples of different parts of the cucumber plants.

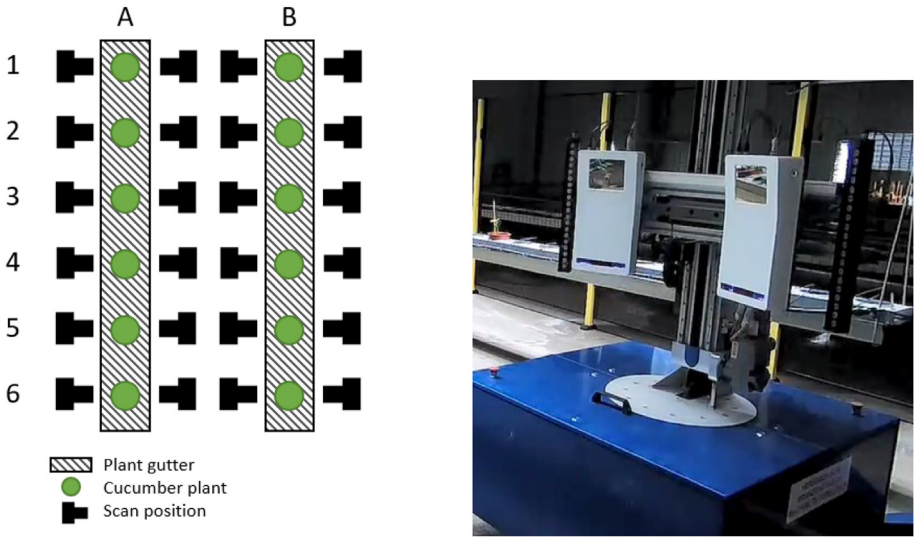

Fig. 2 – Left: Top-view scheme of the 12 plants and the 24 scan positions. The rectangles show plant gutter A and plant gutter B, each containing six plants. Plants are identified as A1 to B6. Right: The point cloud acquisition system, showing the two PlantEye F500 scanners. The mobile blue platform 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.

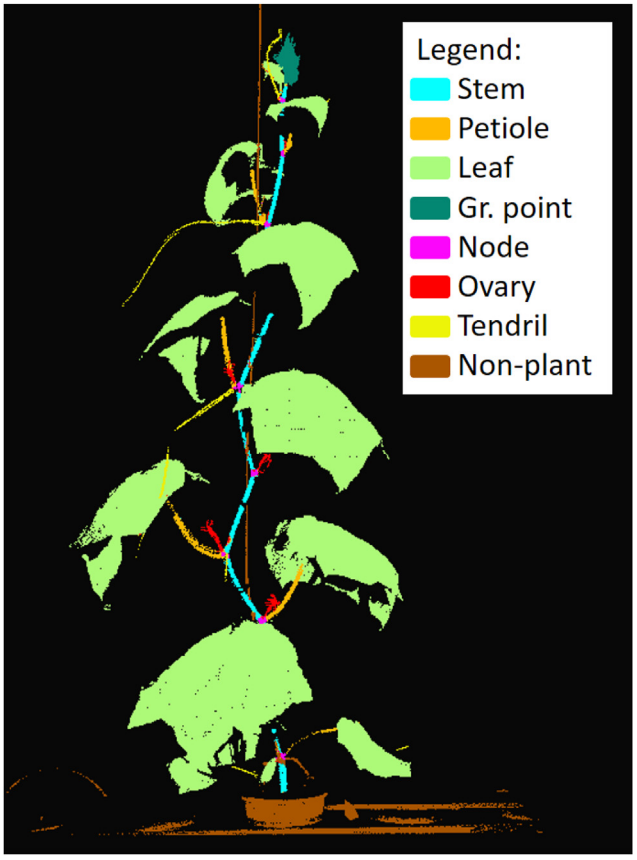

Fig. 3 – Example of a manually segmented point cloud. Note the missing parts in the point cloud.

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 2.4.2.

An overview of the distribution of the different plant parts in the two annotated datasets is shown in Fig. 4. It is clear that the leaf (83.70%) and the non-plant material (9.49%) are over-represented 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.

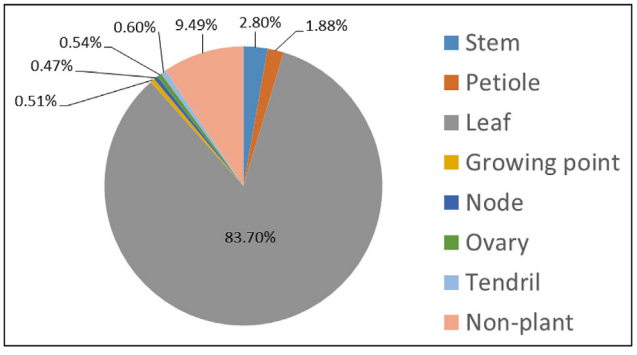

Fig. 4 – Distribution of points over the different plant parts.

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 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 Fig. 2) for these data splits are given in Table 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 3.1. For the other experiments, split 1 was used.

2.2. Point cloud segmentation

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 $2 \times 2 \times 2$ 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 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 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 A1 in the Appendix. An example of a small and a large point cloud and the division in blocks is shown in Fig. 5.

Algorithm 1: pseudo code to divide point cloud into blocks

```
1. split_pc_in_blocks(pc, n):
2.   # pc: original point cloud
3.   # n: number of points per block
4.
5.   N = nr_points(pc)
6.   i0 = 0
7.   i1 = n
8.   block_list = []
9.   pc_sort = sort_on_z_value(pc)
10.
11.  while i1 < N:
12.    new_block = pc_sort[i0:i1]
13.    block_list.append(new_block)
14.    i0 += n/2
15.    i1 += n/2
16.
17.  # Add the last points
18.  last_block = pc_sort[N-n:N]
19.  block_list.append(last_block)
20.  return block_list
```

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

Table 1 – Division of plants over the training, validation and test set for the four cross validation runs.

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

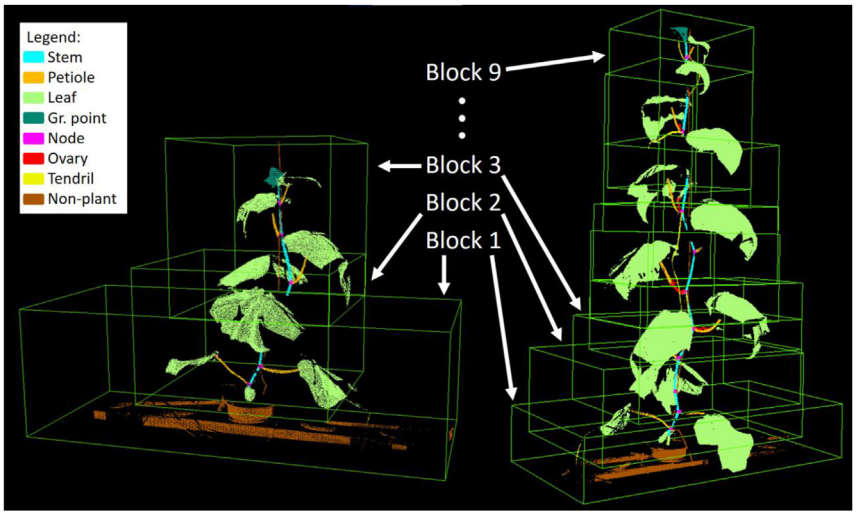

Fig. 5 – Example of two point clouds divided into blocks. The point cloud on the left was obtained on the second measurement day and is divided into three blocks, the point cloud on the right was obtained on the tenth measurement day and is divided into nine blocks. The colours of the points represent 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.

PointNet (Qi et al., 2016), which was one of the pioneering DL 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 $n \times (d + c)$, 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.

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.

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 (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.

$$\mathrm{IoU_i} = \frac{\mathrm{TP_i}}{\mathrm{TP_i} + \mathrm{FP_i} + \mathrm{FN_i}} \quad (1)$$

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, IoU_{micro}, was calculated according to:

$$IoU_{micro} = \frac{\sum_{i=1}^{k} TP_i}{\sum_{i=1}^{k} (TP_i + FP_i + FN_i)} \quad (2)$$

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:

$$IoU_{macro} = \frac{\sum_{i=1}^{k} IoU_i}{k} \quad (3)$$

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 $p < 0.001$.

2.4. Experiments

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

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 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 2.1.4 was done for this experiment. The results are shown in section 3.1.

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 3.2. 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 3.2.

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 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 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

Table 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	X	X	X	X	X	X	X	X
4-class	X		X	X	X			

segmentation task. In section 3.3, the obtained IoU-values and the confusion matrix are presented.

3. Results

In this section, the results of the experiments defined in section 2.4 are presented. First, in section 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 3.2, the intra-observer variability results of experiment 2 are shown. Finally, in section 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.

3.1. *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 Fig. 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.

The confusion matrices for XYZ and XYZRGB are shown in Table 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.

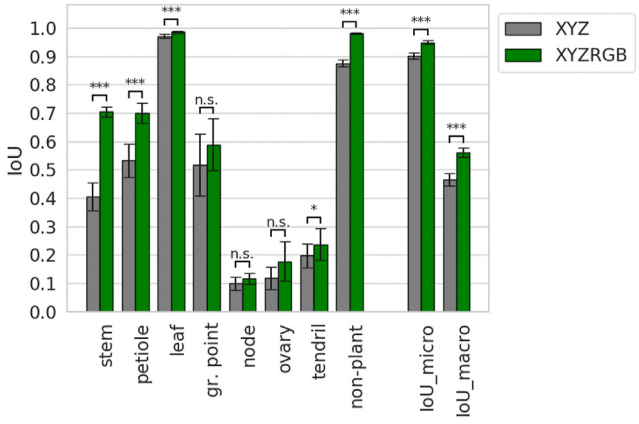

Fig. 6 – Performance on the test set based on only geometric data (XYZ, grey) and geometric and spectral data (XYZRGB, green). The bars give the mean IoU and the error bars show the 95% confidence interval on the mean. The asterisks indicate a significant improvement of the performance when adding spectral data ($p < 0.05$ = *, $p < 0.01$ = **, $p < 0.001$ = *).**

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 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.

3.1.1. *Cross-validation*

The range between minimum and maximum IoU-value for the four cross-validation runs is shown in Fig. 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.

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 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 IoU_{micro} and IoU_{macro}.

3.2. *Intra-observer variability*

As explained in section 2.4.2, the intra-observer variability was measured as the IoU between annotation A and

Table 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**

annotation B. The results are shown in Fig. 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.

Table 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**

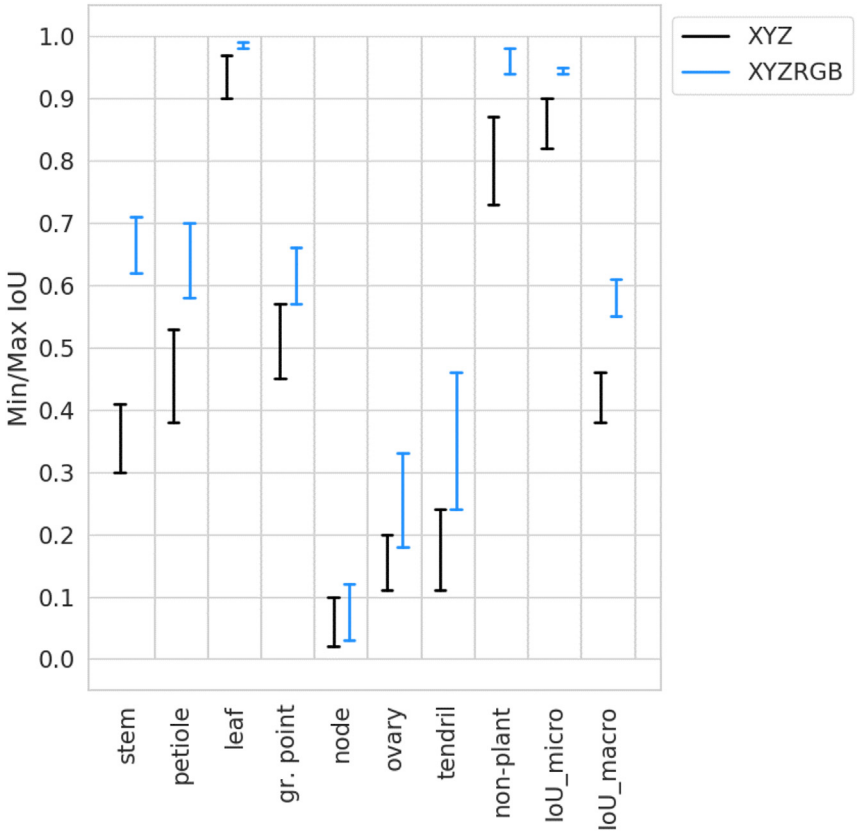

Fig. 7 – Minimum and maximum IoU-values observed over the four cross-validation runs for XYZ (black lines) and XYZRGB (blue lines).

Table 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 Fig. 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	***	***	***	***
IoU_{micro}	***	***	***	***
IoU_{macro}	***	***	***	***

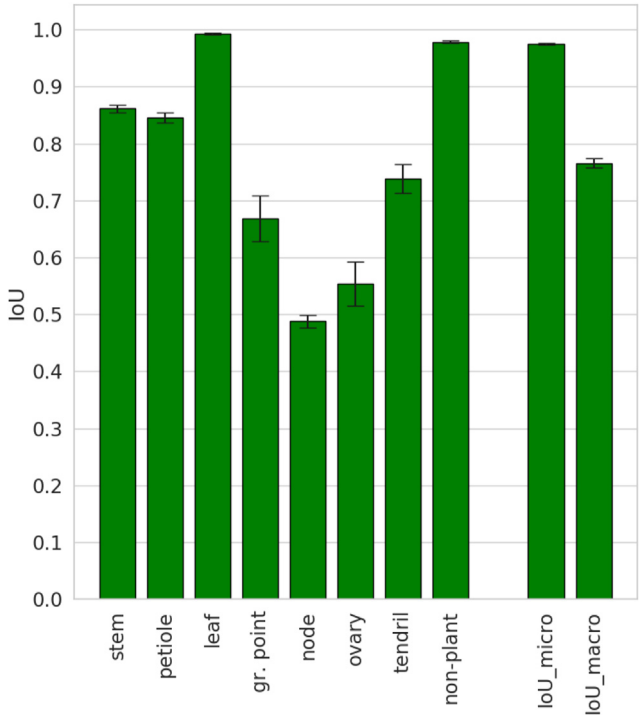

Fig. 8 – Mean IoU between annotation A and annotation B. The bars indicate the mean IoU and the error bars show the 95% confidence interval on the mean. Note that a high IoU-value corresponds with a low intra-observer variability while a low IoU-value corresponds with a high intra-observer variability.

To give more insight in the confusion between classes, the confusion matrix between annotation A and annotation B is shown in Table 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.

3.2.1. *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 Fig. 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.

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 Fig. 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 IoU_{micro}, rising from 0.95 to 0.98 and the IoU_{macro}, rising from 0.74 to 0.86.

The confusion matrix for the 4-class segmentation is shown in Table 7 (for the confusion matrix of the 8-class segmentation, see Table 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%).

4. Discussion and 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 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

Table 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**

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

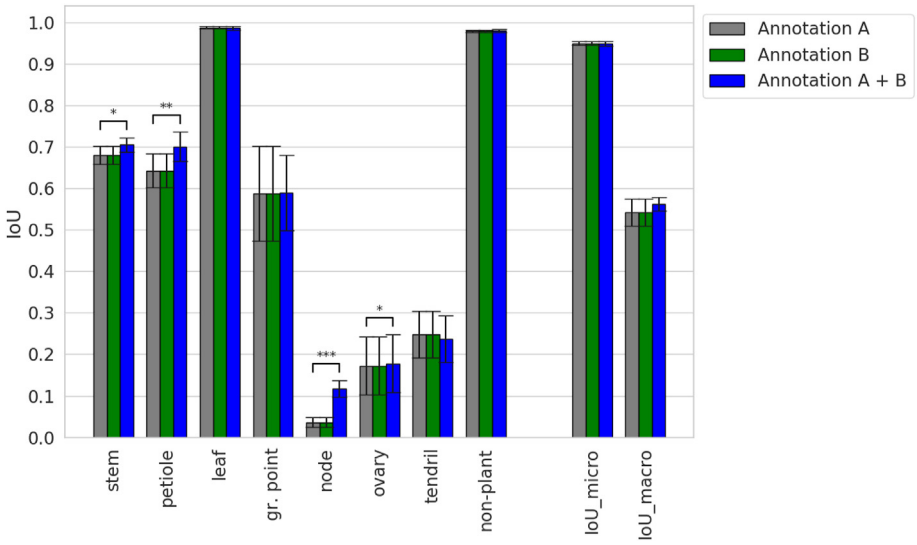

Fig. 9 – IoU for the test set based on training with only annotation A or B and training based on annotation A and B. The bars give the mean IoU and the error bars show the 95% confidence interval on the mean. The asterisks indicate a significant improvement of the performance when trained on two annotation sets as compared to the network trained on one of the datasets ($p < 0.05$ = *, $p < 0.01$ = **, $p < 0.001$ = *).**

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 over-represented 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-

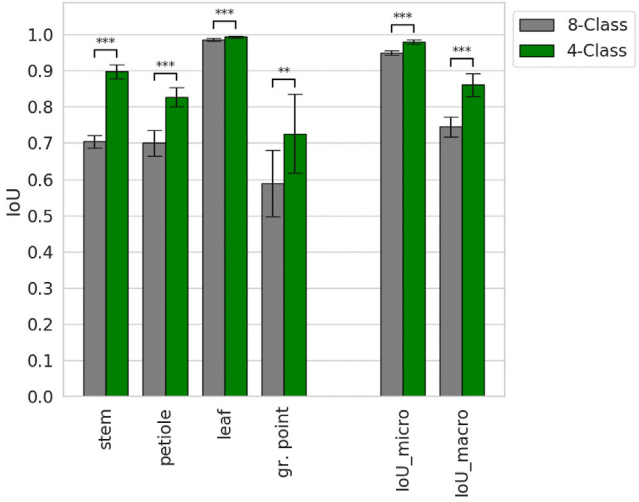

Fig. 10 – Performance on the test set for the original 8-class and the simplified 4-class segmentation task. The bars give the mean IoU for the test set and the error bars show the 95% confidence interval on the mean. The asterisks indicate a significant improvement of the performance for the 4-class segmentation as compared to the 8-class segmentation ($p < 0.05$ = *, $p < 0.01$ = **, $p < 0.001$ = *).**

Table 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**

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 3.2 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.

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 IoU_{micro} and IoU_{macro}. The IoU_{micro} increased from 0.90 to 0.95 and the IoU_{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-operator 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.

Funding

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.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

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

REFERENCES

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

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

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

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

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

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. (2019). SynthCity: A large scale synthetic point cloud. *ArXiv*, 1–6.

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.

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

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

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

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

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

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2016). PointNet: Deep learning on point sets for 3D classification and segmentation. In *2016 fourth international conference on 3d vision (3DV)* (pp. 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.

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

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

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

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

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.

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

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

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