Sensor system red-currant pruning robot

Report on performance of sensors and algorithms

Jochen Hemming[1], Manya Afonso[1], Chrysanthos Papadakis[2], Naftali Slob[2], Bas Boom[2]

1 Wageningen University & Research; 2 IMEC-OnePlanet Research Center

WAGENINGEN
UNIVERSITY & RESEARCH

Referaat

Deze studie beschrijft de ontwikkeling van een mobiel sensorsysteem voor een rode bessen snoeirobot. In het project zijn twee sensorbenaderingen geëvalueerd: sensoren voor metingen vanaf een bewegend platform zoals een LiDAR en camera's om hele rijen planten in kaart te brengen, en een stereocamera op de robotarm voor real-time beeldverwerking. Voor tests binnen in een gecontroleerde omgeving is er een physical twin van rode bessenplanten gemaakt. Handmatige beeldannotatie werd uitgevoerd met behulp van een Virtual Reality-tool. Deep learning-algoritmen werden toegepast voor segmentatie en classificatie. OneFormer3D werd bijvoorbeeld gebruikt voor segmentatie van individuele planten en om onderscheid te maken tussen diverse objecten in de boomgaard, waarbij het classificeren van objecten met een nauwkeurigheid van 85% gaat, het herkennen van individuele struiken gaat met gemiddelde nauwkeurigheid van 60%. Met een MaskRCNN netwerk werden takken van 1 en 2 jaar oud gedetecteerd en geclassificeerd, de ResNet gebaseerde classificatie bereikte een nauwkeurigheid van 83%. De studie benadrukt de noodzaak van een balans tussen sensorkosten, nauwkeurigheid en real-time verwerking.

Abstract

This study explores the development of a mobile sensor system for a red currant pruning robot. The project evaluated two sensing approaches: sensors on a base platform, including a LiDAR and cameras mounted on a moving trolley for full-row scanning, and an end-effector stereo camera on the robotic arm for real-time image processing. A physical twin of red currant plants was constructed for controlled indoor testing. Manual image annotation was enhanced using a Virtual Reality tool. Deep learning algorithms were employed for segmentation and classification. OneFormer3D was used for instance segmentation of individual plants and recognition of the object in the orchard, the object recognition works with 85% object recognition rate, the instance recognition of a plant obtained 60% average precision. MaskRCNN was used to detect and classify 1-year and 2-year-old branches. The ResNet-based classification reached an accuracy of 83%. The study highlights the need for a balance between sensor cost, accuracy, and real-time processing.

Report information

Report WPR-1411
Project number: 4400003799, LVW19.178
DOI: https://doi.org/10.18174/689376

This project was made possible by the contribution of the TKI Topsector Tuinbouw & Uitgangsmaterialen and the Koniklijke Nederlandse Fruittelers Organisatie (NFO) en het Innovatiefonds Hagelunie van Interpolis.

Adress

Wageningen University & Research, Business Unit Greenhouse Horticulture

P.O. Box 20, 2665 ZG Bleiswijk	P.O Box 644, 6700 AP Wageningen
Violierenweg 1, 2665 MV Bleiswijk	Droevendaalsesteeg 1, 6708 PB Wageningen
T +31 (0)317 48 56 06	T +31 (0)317 48 60 01
wur.eu/greenhouse-horticulture	wur.eu/greenhouse-horticulture

Table of contents

Summary

This report presents the findings from the use case 'Sensor System for a Pruning Robot for Red Currant' within the PPP project Handsfree Production in Agri-Food. The goal was to develop a mobile sensor system to measure the 3D structure of red currant plants, a critical component for the robotic pruning process. The cultivation of red currants is labour intensive, and automation is necessary due to rising labour costs and personnel shortages. Unlike apple and pear trees, red currants have a simple geometric structure, making them more suitable for the first steps in the development of a pruning robot.

Two primary sensing concepts were evaluated. The first involved sensors on a base platform, including a LiDAR, colour and Time-of-Flight cameras mounted on a moving trolley for scanning plant structures. These generate an overview of a full plant row resulting in large point clouds. The second concept used an end-effector camera, specifically a stereo camera mounted on the end-effector of the robotic arm. Combined with real-time image processing, this allows the robot to identify and prune branches directly. Additionally, a physical twin of red currant plants was constructed for indoor testing, containing key orchard elements such as crop wires and poles. Data collection was performed in both controlled indoor settings and field conditions.

Manual image annotation was carried out using a Virtual Reality (VR) tool to speed up labelling. The labelled dataset was then used for deep learning model training.

For algorithm development and performance evaluation, different approaches were followed. Instance segmentation using OneFormer3D, a transformer-based model, was employed to segment plants and other structures from point cloud data. The first experimental setup achieved high accuracy, but some misclassifications were observed. A second setup introduced more manual annotations and data augmentation, but segmentation performance declined due to distribution shifts. Deep learning Object detection with MaskRCNN was used to detect and classify 1-year-old and 2-year-old branches. An accuracy of 64% was reached, classification struggled with differentiating similar branch types. Branch classification with ResNet was also evaluated, where various deep learning models were tested. The best accuracy, 83%, was achieved with ResNet18, but classification performance remained imbalanced due to annotation inconsistencies.

The study concludes that sensor selection must balance cost, accuracy, and real-time processing capability. A more robust annotation methodology is needed to improve classification performance, and integrating depth-based segmentation could enhance accuracy. Larger datasets with active learning and synthetic data generation could improve model performance. Further work is required to translate sensor outputs into robotic control commands for precise pruning execution. This study provides valuable insights into the potential and challenges of sensor-based pruning automation, laying the groundwork for future development of robots for pruning.

1 Introduction

This report is the final deliverable of the use case ‘Sensor system for a pruning robot for red currant’ of the public-private-partnership (PPP) project Handsfree Production in agri-food (LVW19.178). The aim of this use case was to develop a mobile sensor system for measuring 3D plant structure (initially for red currant plants). This activity can be seen as an extension of the parallel ongoing PPP The Next Fruit 4.0.
In The Next Fruit 4.0, a multifunctional orchard robot is being worked on, one of the envisioned tasks of the robot is pruning of fruit trees and soft fruit bushes such as red currant.

The cultivation of red currants is labor-intensive, especially because pruning and harvesting is still purely manual labor. Because labor costs continue to rise and because there is a threat of a shortage of qualified personnel, soft-fruit growers are looking for technological alternatives to fill in these activities and thus make an efficiency gain in their business operations. Compared to apple and pear trees, the geometrical structure of red currants is simpler. This means it is expected that a pruning robot for this crop can be realized more quickly. There are several initiatives in the field of pruning robots worldwide, but these have all stopped or are in an early experimental stage. Robust mapping of the 3D plant/tree structure with sensors and determination of the correct locations for pruning cuts are the biggest challenges in all these initiatives. To achieve a fully autonomous and robust pruning robot solution, it is necessary to develop a sensor system that can also recognize thin (side) branches, as well as obstacles for the robot such as crop wires, poles and irrigation pipes and/or drippers. Being able to distinguish individual plants (which branches belong to which plant) is also essential in order to determine how many and which branches need to be pruned.

2 Research and selection of suitable sensor candidates

Based on results from previous projects and on literature and market research two basic sensing concepts were further investigated. The first concept uses a setup of a high-resolution LiDAR and cameras that are mounted on a moving base platform (or in the future possibly on the vehicle with the robotic arm). These sensors are used to scan the plant structure from a distance. Processing of the data is done offline after scanning, resulting in large point clouds/image data sets representing a full row of plants. To be able to use this information to control the pruning robot, proper geo-referencing and position calibration of sensors and robot is needed such that the robot can execute the pruning tasks at the correct plant on a later moment.

The second concept uses a camera system mounted on the robotic end-effector. This local camera can actively be positioned with the robotic arm. By preference the camera should not only deliver colour images but also 3d information, i.e. using stereo-vision. Processing of the images from this camera is supposed to be done in real-time, allowing to use the information to directly control the robotic arm with the pruning tool. In that case only a local hand-eye calibration is needed to execute the pruning tasks at the correct location.

2.1 Sensors on a base platform

Within this project, IMEC used two sensor setups to record the red currant bushes both indoor and outdoor. The first setup is shown in Figure 1, which contains both a 360 degree LiDAR from Leica (BLK ARC Imaging Laser Scanner, around € 40 000), two Time of Flight cameras, one from Microsoft (Azure Kinect, € 600) and one from Lucid (Helios 2, € 2 500) and the Imec Hyperspectral Camera (around € 25 000). Within this research, the focus was on depth sensing, thus we ignored the data generated by the Hyperspectral Camera. Still, the hyperspectral data might be valuable for future work. In our research, both the data from the LiDAR and the depth camera (our preference is the Azure Kinect because it price and the data quality) were useful for the analysis of the trees/attributes/branch age. However, for both the LiDAR and Azure Kinect, we encountered problems that we were partly able to solve but should be further investigated in the future. The first problem is that the point resolution of the Leica LiDAR is not very dense, where the number of points per branch is often the limiting factor for the recognition. The second problem is that the colouring from the cameras that are aligned with the LiDAR does not perform well for tree objects. Although this is a well know problem the resulting colour data can limit the recognition process since background colours are mistakenly being used. For the Azure Kinect, the resolution of the depth maps is high compared to the LiDAR. Combining the point cloud is however more difficult and solved partly by alignment of the depth maps and LiDAR scans. The Azure Kinect gives a high resolution RGB image as well as a lower resolution depth image, but this results in colouring average where depth pixels do not always have the correct colour. The data of the Azure Kinect can be analysed both by image recognition and point-cloud recognition algorithms, although the alignment between different frames recorded by the Azure Kinect is difficult.

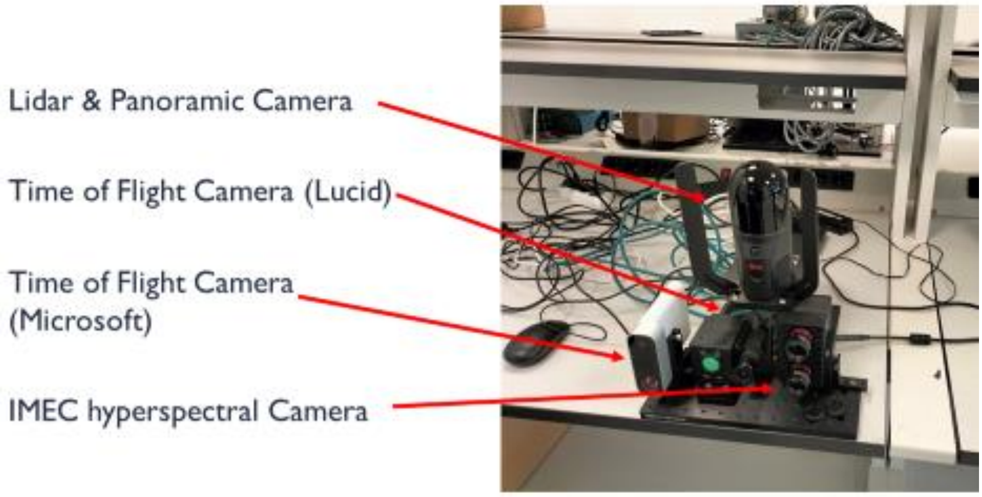

Figure 1 *First base sensor platform that combines multiple RGB camera's and depth sensors together.*

The second sensor system is developed by IPI (https://ipi.ugent.be/) which is a research group at the University of Gent. The setup consists of 2 high resolution (1920 x 1200 pixel, depending on frame rate settings) RGB cameras (€ 2 000) combined with an Ouster LiDAR (OS1-128 with 128 beams), that costs about € 20 000. IPI created its own SLAM (Simultaneous localization and mapping) implementation based on the LiDAR, allowing the sensor setup to determine both the position of the LiDAR and the 2 cameras. Based on the camera's, stereo vision is performed to calculate the depth. Instead of performing this on 2 images, the method IPI used is combining images from all the camera positions taken while moving the camera. The technology behind this is called gaussian splatting, first introduced by Kerbl et al. (2023). By sampling the Gaussian kernels, 3d point clouds can be generated. The disadvantage of this technology is that it is computationally expensive and is currently only possible to compute offline on a GPU power computer where it might take around 30 minutes to complete for an entire scene.

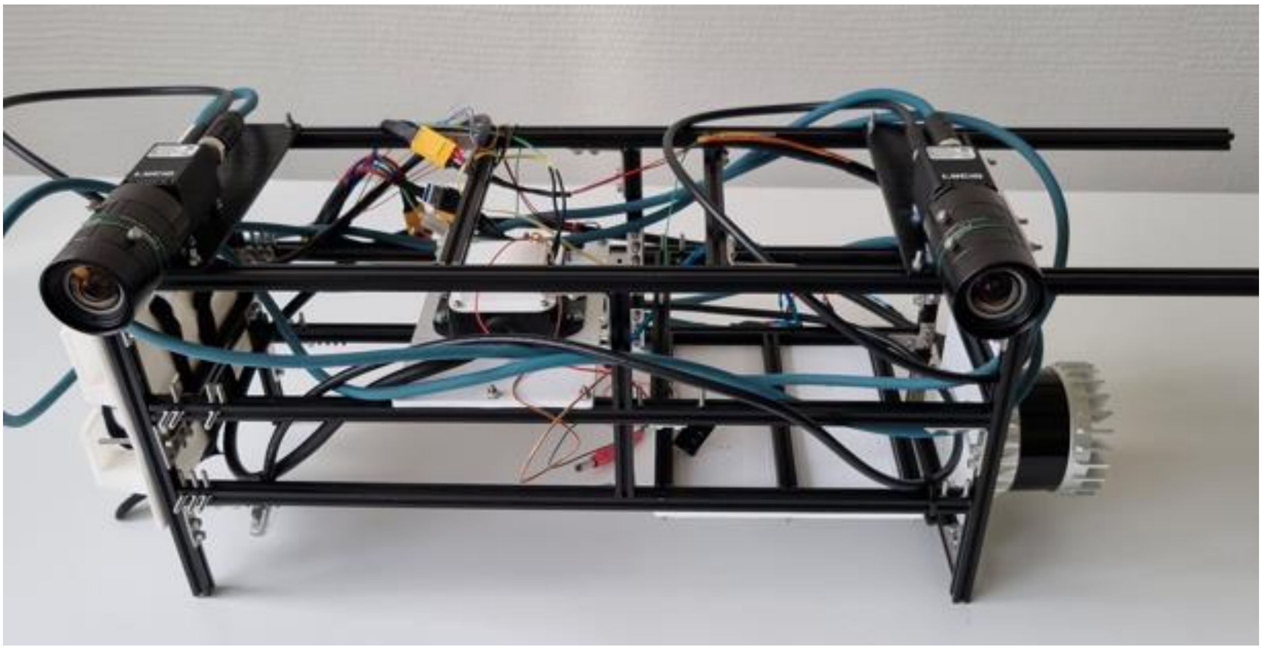

Figure 2 *Second base platform sensor that combines two high resolution RGB cameras and LiDAR used in automotive.*

2.2 Camera on the end-effector

The purpose of a camera on the end-effector is to be able to acquire image data on a short distance to the target and therefore with high spatial resolution. In the course of this project different camera candidates have been studied for their suitability. These included different models of Intel ReaSense cameras (D405, D455, L515), a Stereolabs ZED-X mini camera and a PhotonicSense APICube[1]. Finally, an Intel RealSense D405 stereo camera was selected. The price of this camera is about € 290. The camera is mounted on the end-effector as shown in Figure 3 and is small and lightweight (4.2 cm × 4.2 cm × 2.3 cm; 60 g). It provides colour and depth images between 7 cm and 50 cm distance. This makes this camera suitable for detecting detailed features or improving the estimation of the target position based on additional sensors on the vehicle where the robot arm is mounted.

Figure 3 *Pruning end-effector with top-mounted Intel RealSense D405 camera.*

[1] https://photonicsens.com/

3 Physical twin with red currant plants for indoor use

To be able to develop and test sensors and algorithms independently of the season, a physical twin with real but dead red currant plants was constructed. For reference, Figure 4 shows photos from the orchard during the growing season and during the dormant season. The major pruning activities are carried out in the dormant season. The physical twin that is shown in Figure 5 and Figure 6 contains all the elements that are also present in the orchard, next to the plants also poles, crop wires, clips, irrigation drippers.

Figure 4 *Red currant plants with leaves and fruit during summertime (left photo) and during wintertime (right photo).*

Figure 5 *Photos taking during the construction of the physical twin of a red currant plant row.*

Figure 6 *The physical twin of a red currant plant row as used during indoor experiments.*

4 Data collection and data annotation

4.1 Sensors on a base platform

On multiple days, data with the sensor systems has been collected, both indoors from the physical twin and outdoors at the Fruit Research Centre of WUR in Randwijk (Figure 7). Data collection took place in autumn and winter of 2024. A typical coloured point cloud example of a dataset that is collected is visualized in Figure 8.

Figure 7 *Data collection with the global sensor setup in the orchard.*

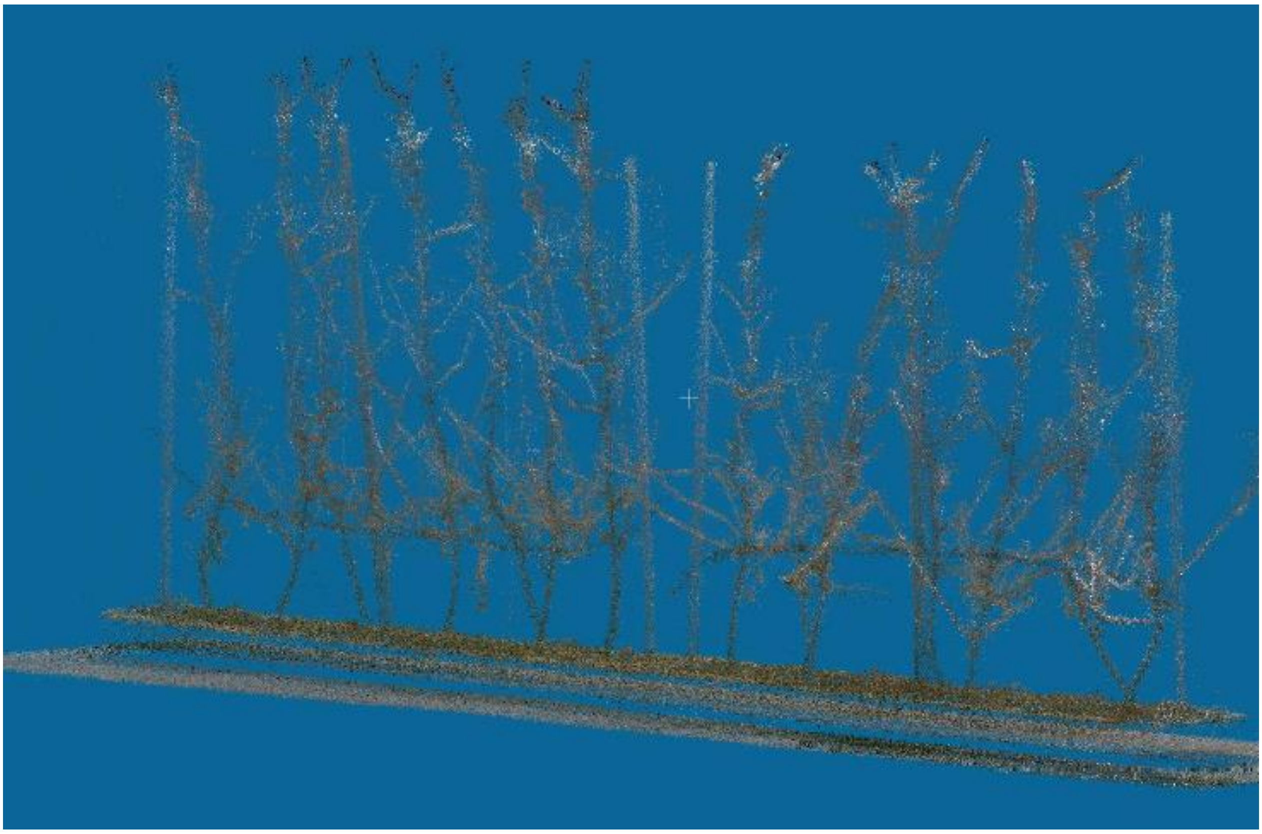

Figure 8 *Colored LiDAR point cloud data from the Leica BLK ARC of the indoor Red Currant plants.*

In Figure 9, an example of the output of the IPI Gent setup is shown. This is however dependent on GPU processing and on taking correct measurements. One of the problems is the field of view of the camera for which it was difficult to capture the entire trees while driving to the orchard. In the future, this technology can become potential interesting however currently we struggle a lot with the exact camera and LiDAR registration to obtain enough data to train our deep learning algorithms on. For this reason, we decided to continue for the remaining part of this study with data from the Leica LiDAR setup.

Figure 9 *High resolution point cloud from the setup of IPI Gent showing the potential of high resolution. This however requires GPU processing taking 4-8 hours per scene which contains around 5-8 plants.*

To be able to train a deep-learning network to distinguish between the different plant parts and objects in the scene (i.e. plants, poles, dripline, wires, ground) training data needs to be annotated. In a first attempt, deep learning models were trained with manual annotations from apple tree data to segment the different parts of a red currant crop. This however failed and the decision was taken to annotate red currant data for this purpose. To speed up the manual annotation time an IMEC developed annotation tool is uses that allows in Virtual Reallity (VR) to interact with the point clouds. A schematic representation of the annotation tool is shown Figure 10. The VR tool is similar to (Garrido et al, 2017) allowing users to view point clouds in VR, interact with points using tracked controllers, and add annotations manually with a sphere brush.

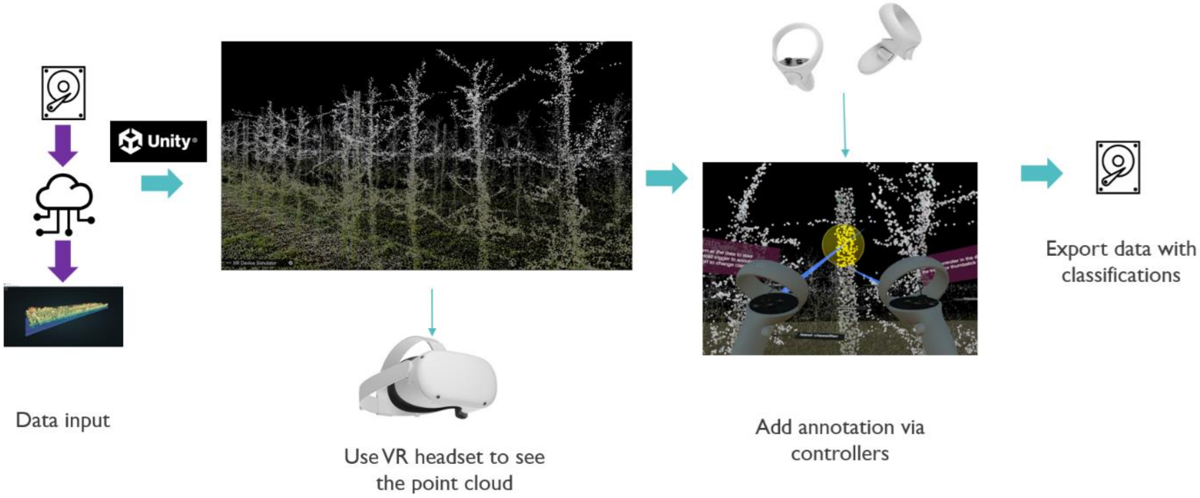

Figure 10 *Annotation tool for 3D point cloud allowing user to perform annotation in 3D using the VR headset.*

The annotation tool starts with importing and visualizing point cloud data in Unity. There is the development of manual annotation algorithm in Unity and the use of software (Cloud Compare and PyCharm) and the tools necessary around the annotation tool to make sure the data is in the correct format. The data is later export toward the machine learning algorithms in a format that can processed. The VR tool's visualization performance is enhanced by using octree structures, circle textures, and Level of Detail (LOD) techniques to improve rendering speed and spatial comprehension. The tool also supports real-time colour changes for selected points, enhancing the intuitiveness of the annotation process. The tool was tested in a within-subjects study, showing significant time savings in point cloud annotation compared to traditional desktop-based software. An example result of the annotation process can be seen in Figure 11.

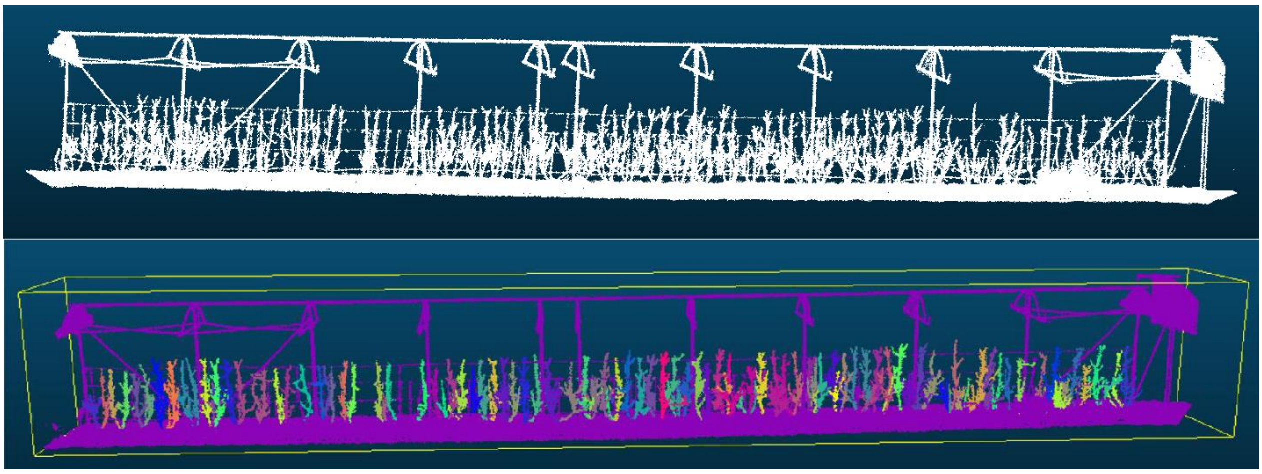

Figure 11 *The original LiDAR scan without colour projection is shown above, below the annotated LiDAR scan. Each colour represents a unique plant (instance). Purple colour represents background objects that do not belong to the 'plant' category ('rest' class).*

4.2 Camera on the end-effector

The approach with the camera on the end-effector is to (1) acquire short distance high resolution images from a plant, and (2) to use this information to make the robotic arm follow the main leader of a plant. While following the main leader, side-branches should be identified and classified (1-year old wood, 2-year old wood).

Multiple image datasets were collected in the orchard with different camera candidates as described above. Some example images and annotations are shown in Figure 12. From this data, the RealSense D405 and ZED-X mini camera had the best resolution to allow noticeable differences for annotating the one year or two year old branches. Further, only the outdoor colour images were selected. To annotate the data the Darwin v7[2] tool was used, with Meta's Segment Anything Model[3] (SAM) used as the backend. This tool allows hovering with the mouse over a branch in the image and the model detects a branch instance based on the pre-trained model. This way instance masks for 3 classes – 1 year branch, 2 year branch, and trunk, were annotated. The criterion for defining an object as a 1 year or 2 year branch was done by manual visual judgment, and based on the 1 year branches being lighter and smoother. This meant that Darwin/SAM's ability to detect a branch mask depended on whether the camera was close enough to the tree for proper branch sizing and sufficient illumination. In cases where branches were very thin, the images from the global sensor platform and the other end-effector cameras did not always detect branches.

A total of 95 images from the RealSense D405 and 34 images from the ZED were annotated with instances for trunk, 1 year and 2 year branches. Such a dataset can be exported from Darwin to the COCO or YOLO format to train deep-learning models, in this case for object detection. By applying each branch's mask on the RGB image and cropping to its bounding box, the individual branch images for classification were prepared.

[2] https://darwin.v7labs.com
[3] https://segment-anything.com/

Figure 12 *Example images with Darwin annotations. (top left) Azure Kinect camera of IMEC platform; (top right) ZED X-mini camera; (bottom left) RealSense D455 vertically oriented; (bottom right) RealSense D405 horizontally oriented.*

Figure 13 *Examples of annotated masks: red: trunk, blue: 1 year old branch, green: 2 year old branch.*

A similar approach was used for preparing more data for classification by applying an optical flow-based foreground trunk and branch segmentation. For details about this method is referred to the publication of You et.al. (2022). This method is referred to as the “follow-the-leader method”. This data was collected in the Randwijk orchard in December 2024 by moving the robot arm with the pruning end-effector and mounted RealSense D405 camera with constant velocity upwards along a main leader of a red currant plant while acquiring the image stream (Figure 14). The side branches were annotated as 1 year or 2 year old, by visual inspection. A reject class was also included as some spurious objects such as pipes may get misclassified as branches.

Figure 14 *Example of a RGB image acquired in December 2024, with the branch segmentation results of follow the leader method.*

The final dataset from the camera on the end-effector used for training is summarized in Table 1.

Table 1 *Final image dataset of end-effector camera for the classification.*

Dataset	**#images**	**#1 year**	**#2 year**	**#trunk**	**#spurious**
Darwin	129	874	230	310	0
Follow leader	441	335	107	0	380

5 Development of sensing algorithms and performance evaluation

As described in the previous section data from two different sensor systems was collected. Data from sensors on a base platform and data from the cameras on the end-effector. Different research questions have been addressed with this data.

With data from the sensors on a base platform it was investigated if it is possible to create a full overview of a (part of) a plant row. Also it was investigated if it is possible to use that data to identify the different plant parts and objects in the scene. Next to the plants the algorithm should be able to identify also crop wires, poles and irrigation drippers. Also it was investigated if it is possible to distinguish individual plants (which branches belong to which plant). This approach is further detailed in section 5.1.

With images from the camera on the end-effector it was investigated if it is possible to develop a real-time object detection model that is able to distinguish between different types of side branches (1-year-old and 2-year-old wood). With this information it is possible to make a decision which branches should pruned.

5.1 Instance Segmentation with OneFormer3D

5.1.1 Methodology

With data from the sensors on the base platform an instance segmentation was trained. The selected approach to solve the task described above is OneFormer3D (Kolodiazhnyi et al, 2024), a state-of-the-art transformer-based (Vaswani et al, 2017) deep learning model that is suitable for instance segmentation on 3d point clouds. There are three main segmentation tasks on 3d point clouds, namely semantic, instance and panoptic segmentation. Semantic segmentation aims for the extraction of objects in a scene which belong to the same category, e.g. red currant bushes, whereas instance segmentation takes this task one step further and aims for the distinction of individual objects in each class. Panoptic segmentation is the combination of semantic and instance segmentation. Oneformer3d is the first multi-task unified 3d segmentation framework that can achieve state-of-the-art results in all three tasks using a single end-to-end trainable model.

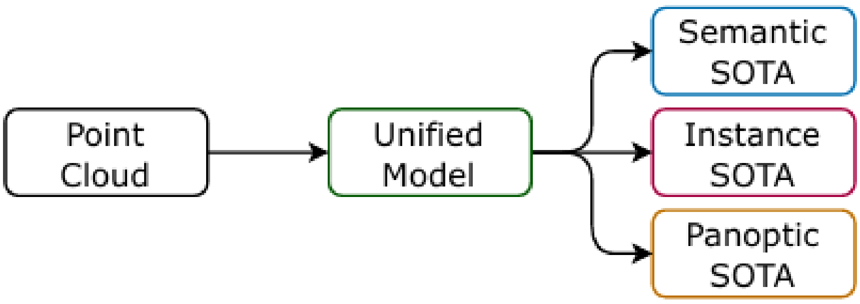

Figure 15 *Oneformer3d unified approach to both Semantic, Instance and Panoptic Segmentation (original figure in https://arxiv.org/pdf/2311.14405).*

The main OneFormer3D framework accepts a raw 3d point cloud as input and after feature extraction for each 3d point. Small clusters will be formed each represented by a feature descriptor. Those descriptors will then be used as input to the transformer layer of the model which is responsible for decoding the descriptors into some meaningful information. Finally, the output of the transformer gets subjected to some operations that bring it to an inferable data format.

5.1.2 Experiments and Results

The OneFormer3D should be able to predict both the different classes (wire, pole, frame, plant) which is called semantic segmentation and the individual plant (or bushes) in the rows which is called instance segmentation. In this research, we first focus on the instance segmentation, since this was the biggest unknown. Multiple outdoor and indoor scans were recorded where a scan contains a row of red currant bushes. During data collection, the Leica LiDAR often went into an error state due to a defect later fixed by the manufacturer. For this reason, some scans contain only a part of the row of plants. For all the experiments on instance and semantic segmentation using OneFormer3D the scans from the orchard were split into training and test data. For the training data, a full outdoor row, and another part of a different outdoor row were used along with half of the indoor scan. The scans contain a large number of points, so for that reason, the scans are automatically split by using an adjustable selection window for the bushes in the scan. Each window contains a random number of bushes (instances) to help the model generalize and perform better in more complex outdoor scans. The maximum size of the window was set to contain 7 bushes to avoid memory overload. A few examples of smaller scenes are provided in Figure 16 and Figure 17.

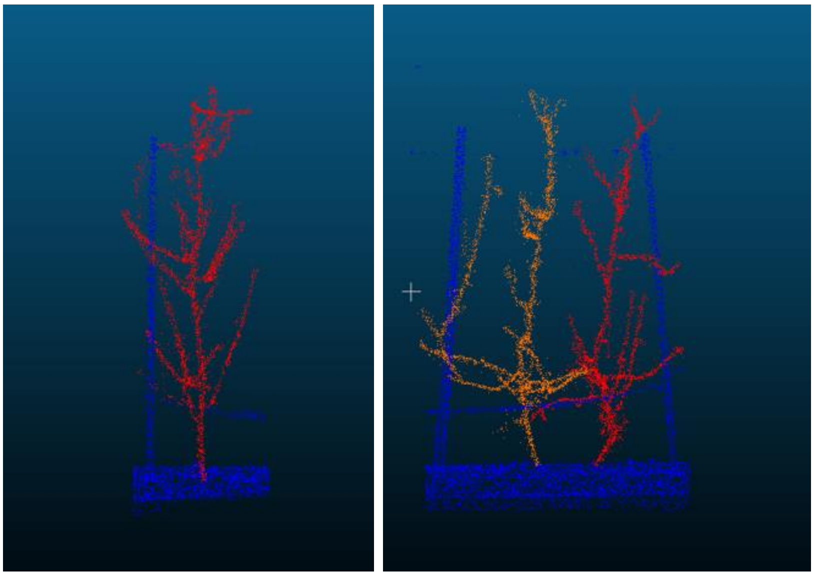

Figure 16 *Different scenes of the indoor scan with adjustable window size. Left the size of the window was set to '1' so it includes only one instance whereas on the right the window size is equal to '2'.*

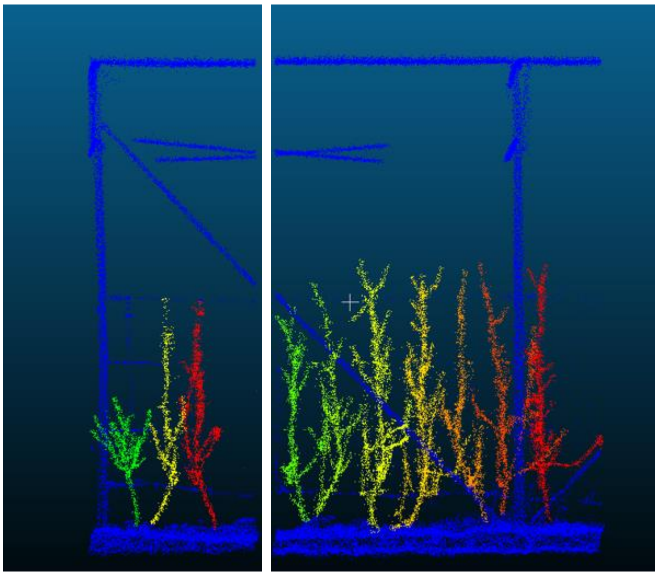

Figure 17 *Different scenes of the first outdoor scan with a selection window that was set to contain multiple instances.*

First experimental setup
For the first experiment the model was trained for 256 epochs with voxel size equal to 0.03 and a test set which consists of 5 unseen scenes was used. The inference results are presented in Table 2 and Table 3, for the semantic and instance segmentation tasks respectively. The precision and recall scores were calculated for an Intersection over Union (IoU) > 0.50. The semantic table displays the IoU of tree and rest class and the accuracy (acc) score is calculated per-point, for the overall prediction and each class individually. Qualitative performance results of the model are depicted in Figure 18.

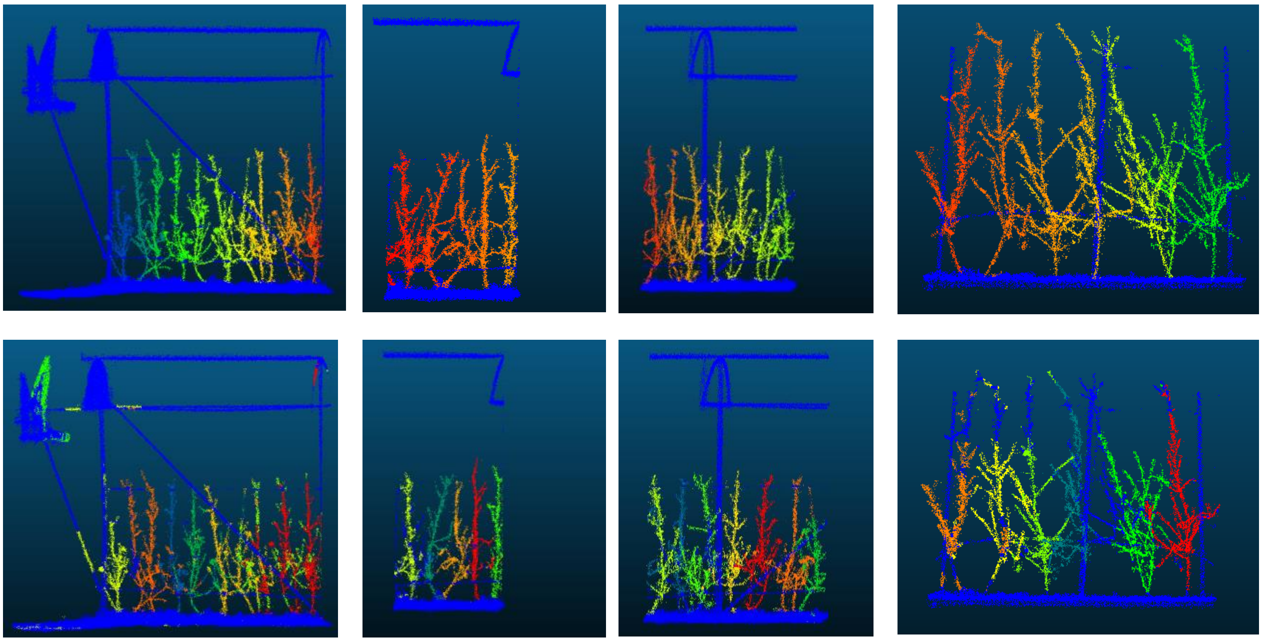

Figure 18 *Results of the segmentation model. Top row are the scans with ground truth labels where each color represents an actual class of the scan. Bottom row are the inference results of the model. Blue represents the 'rest' class meaning background objects such as poles or supporting wires.*

Table 2 *Semantic segmentation results for each class.*

classes	accuracy (tree)	accuracy (rest)	mean intersect over union	accuracy	class accuracy
results	0.90	0.96	0.93	0.97	0.97

Table 3 *Results and metrics that were used to evaluate semantic segmentation (AP=average precision, IoU=intersection over union).*

Classes	AP (IoU = 0.25)	AP (IoU = 0.5)	AP (IoU = 0.5-095)	Precision (IoU = 0.50)	Recall (IoU = 0.50)
Tree	0.75	0.70	0.61	1.00	0.70
Rest	1.00	1.00	0.97	1.00	1.00
Overall	0.88	0.85	0.79	1.00	0.85

As can be seen from the tables above, the model achieves high accuracy on the semantic segmentation task with an IoU 0.9019 and 0.9604, for the 'tree' and the 'rest' semantic classes, respectively. For the instance segmentation task, the overall average precision (AP) for the tree class (calculated from 0.50 to 0.90 with a step of 0.05) suggests that the model is able to perform relatively good but with still a lot of small inaccuracies in the final predictions. Those inaccuracies can be seen in Figure 18, where the model misclassifies some background points as belonging to one instance. For example, in the most-left inference result the top of the arch is colored as green, meaning that the model categorizes that part as an instance.

Furthermore, a lot of instances are grouped together as one instance (bushes that are the same color in the inference row). That could be due to the max size of the selection window set to 7, the model is biased towards outputting that maximum number of instances. Also, it can be seen in the right-most inference result that the tops of the bushes are classified as part of the 'rest' class. The reason for this is probably that, that scene belongs to the indoor scan but the model was trained mostly to outdoor scan scenes (due to lack of data). This makes it more biased towards outdoor scenes. To boost the model's performance for generalizing larger scenes a second experiment was conducted.

Second experimental setup

For the second experiment, further manual annotation on scans together with various data augmentation techniques are used to increase the model's segmentation accuracy. Various new scenes are created that contain more sparse rows of bushes and clouds for each class individually, namely bushes and the rest class. Figure 19 below depicts such scenes that is used in the training set. Additionally, the outdoor scans were cleaned be removing a subset of points. Thus, points that are higher than the bushes, such as tops of long poles and arches, are removed along with some parts of the ground introducing a more balanced scan with relatively even number of points between the two classes (Figure 20).

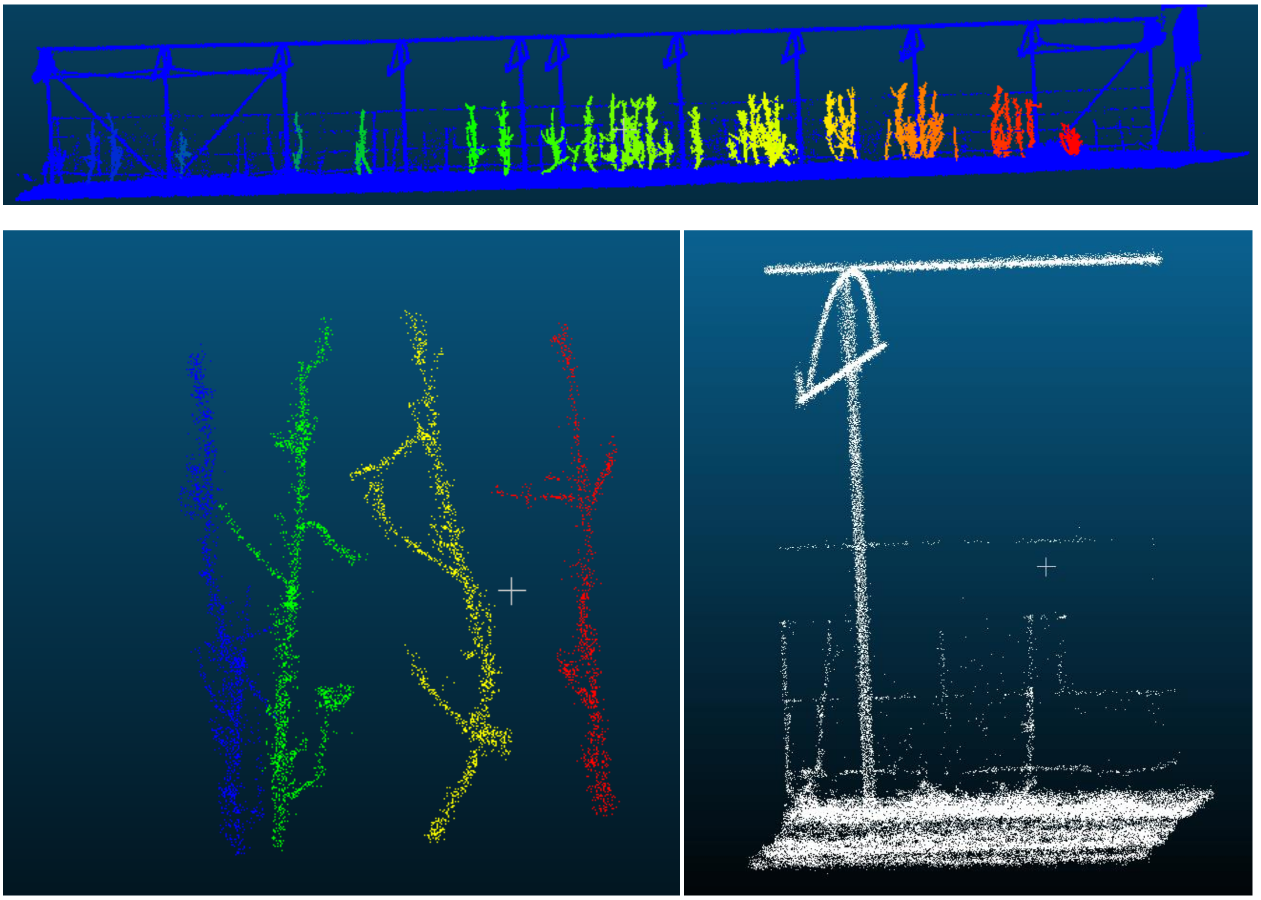

Figure 19 *Examples of scenes in the training set. Top is a row of more sparse bushes. Bottom-left is an only-bushes scene and bottom-right is an only-rest class scene.*

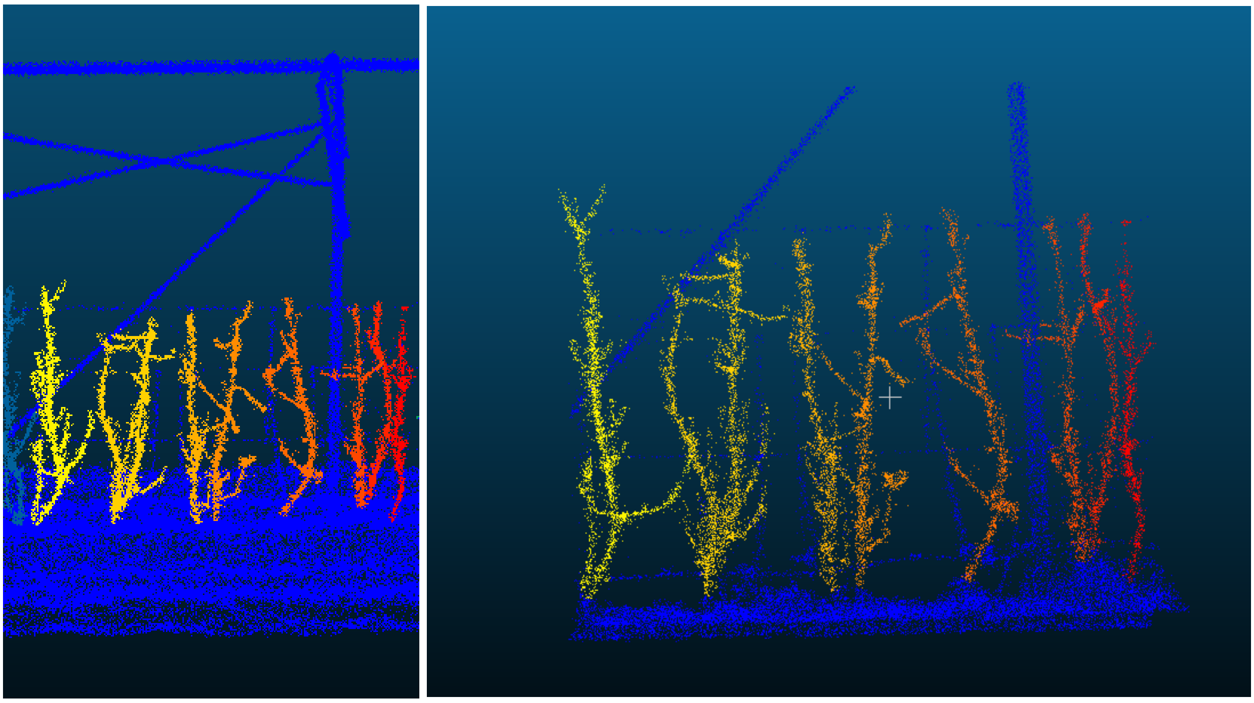

Figure 20 *More augmented training examples. Right is a filtered scan with less ground points and partially removed background objects compared to the original scan on the left.*

Following these procedures, both the original and augmented scenes were used during training and the model was trained for 256 epochs with a voxel size of 0.03 and the qualitative results of the second experiment are depicted in the following figures.

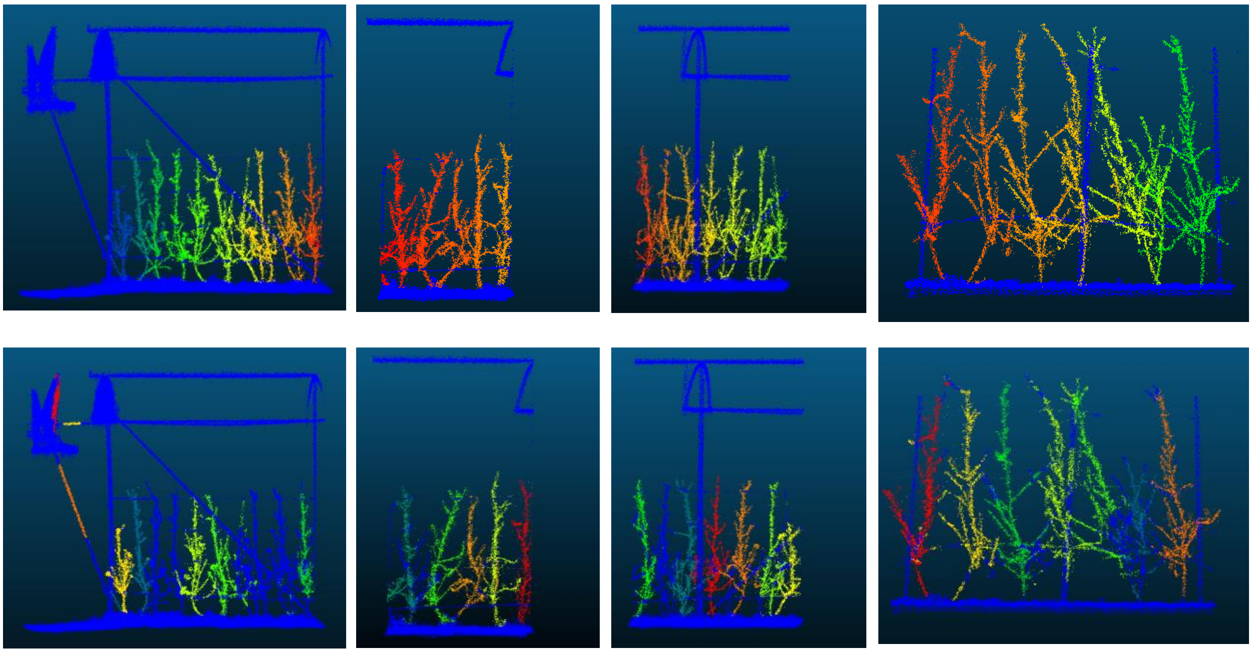

Figure 21 *Results of the model's performance for the second experiment. The top row shows the ground truth scans and the bottom row are the inference results. It can clearly be seen from the left scan that the model failed to segment several bushes from the background objects and misclassified parts of the background objects as instances (coloured arches).*

Table 4 *Second experiment semantic segmentation results.*

Classes	accuracy (tree)	accuracy (rest)	mean intersect over union	accuracy	class accuracy
Results	0.87	0.95	0.91	0.96	0.96

Table 5 *Results and metrics that were used to evaluate semantic segmentation.*

Classes	AP (IoU = 0.25)	AP (IoU = 0.5)	AP (IoU = 0.5-095)	Precision (IoU = 0.50)	Recall (IoU = 0.50)
tree	0.61	0.49	0.20	0.90	0.55
rest	1.00	1.00	0.94	1.00	1.00
Overall	0.81	0.74	0.57	0.95	0.77

The AP of the tree class in Table 5 indicates that the model performs worse than in the first experiment. It seems data augmentation techniques, even though introducing more training data, didn't result in the model to improve generalization for different cases resulting in inaccurate segmentation. That's probably because since more data were introduced the model had to adapt to scenes with completely different class-points distributions.

Third experimental setup
In this experimental setup, the class segmentation of poles, ground, wires and frame is combined with the instance segmentation algorithm. Our assumption is that the neural network might find it easier to separate these classes, since they have a distinct appearance. This will also give robot a better understanding of the scene where interacting with any of these objects will give problems.

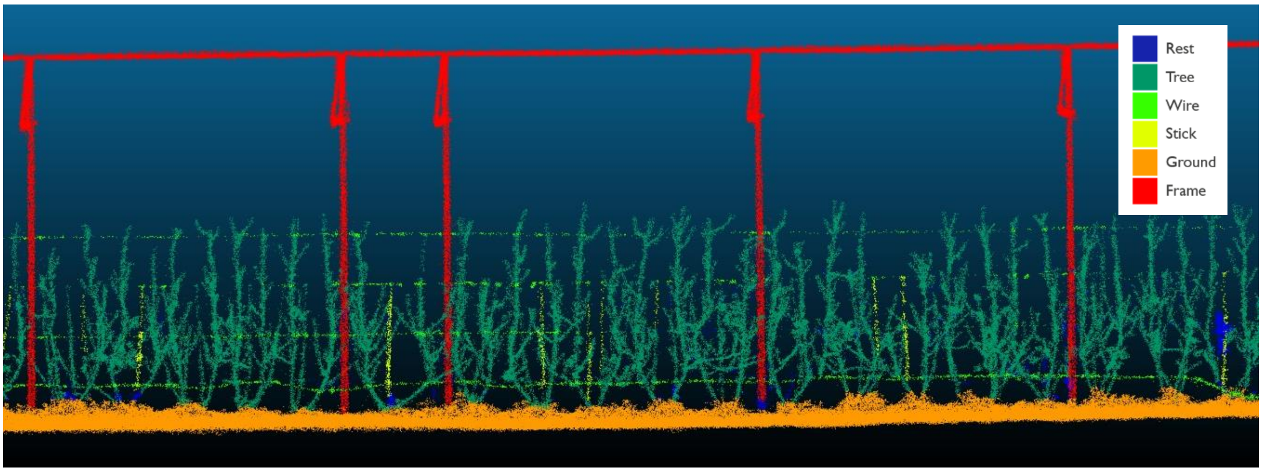

Figure 22 *An annotated outdoor scene where every label contains a different class and shown by a different colour. Classes are described in the legend.*

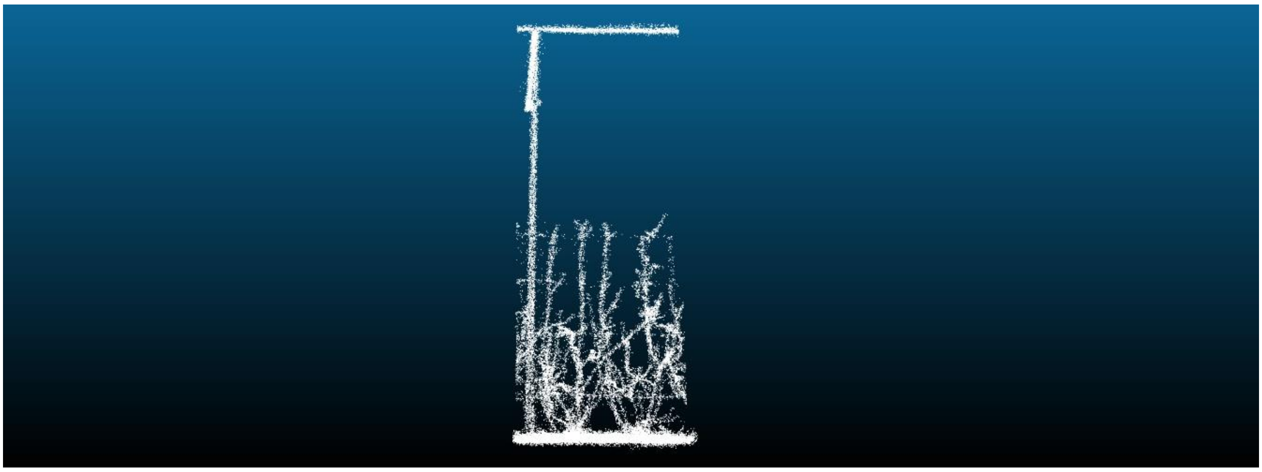

Figure 23 *Since the entire point cloud cannot be processed at once, the point cloud was cut into smaller scenes containing only a smaller subset of plants which can processed by our deep learning approach.*

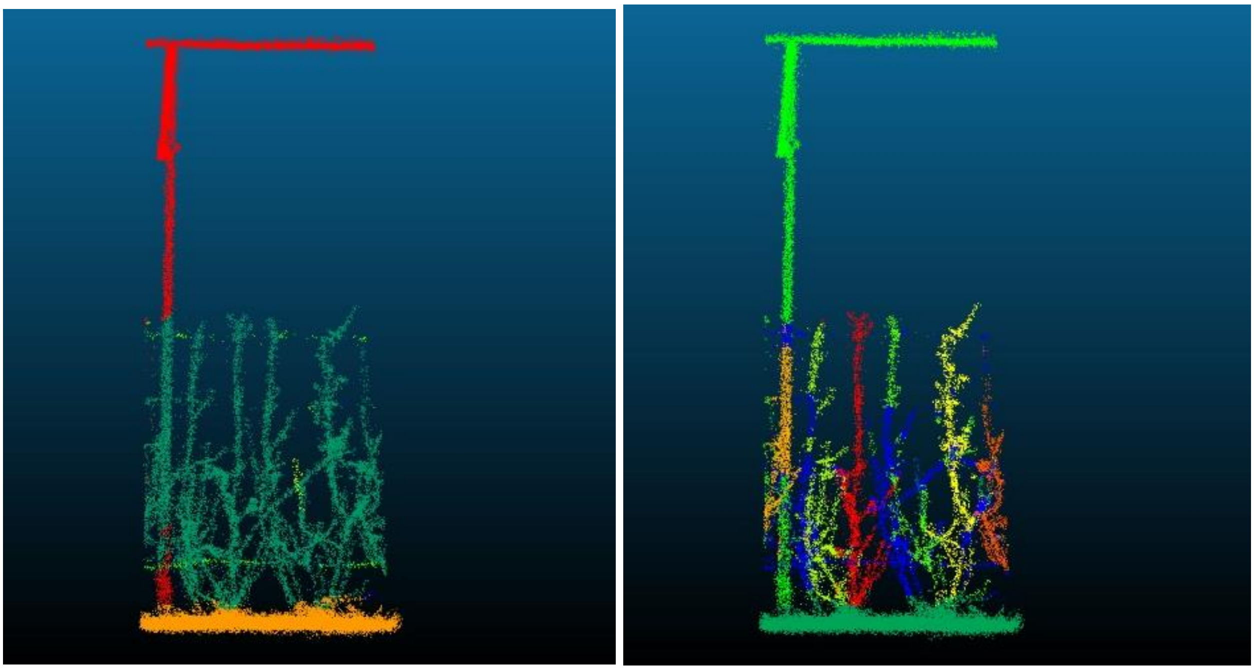

Figure 24 *Example results of both the semantic (left) and instance (right) segmentation are shown. The segmentation results are promising however the instance segmentation are still not as good as expected.*

Figure 22 shows the annotation of all the different objects in the scene for an entire row. Figure 23 shows an typical input point cloud, where a smaller part of the point cloud is used to calculate the inference. Figure 24 shows the qualitative results of both the semantic and instance segmentation of the plants where the semantic segmentation seems to work, although still making a big mistake at the pole, the instance segmentation results are worse than in the first experiment. For OneFormer3D to compute a scene containing around 5 trees as shown in Figure 23, it will take a couple of seconds for a computer with GPU abilities.

Table 6 *The accuracy per class together with the mean intersection over union (miou) for all classes and the point accuracy and mean accuracy per class.*

Classes	rest	tree	wire	stick	ground	frame	mean intersect over union	accuracy	class accuracy
Results	0.55	0.96	0.69	0.78	0.99	0.99	0.83	0.99	0.87

Table 7 *Average Precision of only the tree class, instance segmentation was not activated for the other classes.*

classes	AP (IoU = 0.25)	AP (IoU = 0.50)	AP (IoU = 0.5-095)	Precision (IoU = 0.50)	Recall (IoU = 0.50)
tree	0.64	0.57	0.48	1.00	0.57

In Table 6, the qualitative performance of the semantic classification looks promising. In Figure 24 (left) it can be noticed that some points of the pole are misclassified, however this is only a very small percentage of all point in the scene. For this reason, a very high accuracy is calculated, however in practise on the most difficult points it made these mistakes. We observed that the class wire and stick are also difficult (Table 6), although for wire it is not always clear which points belong to the wire and tree as they intersect with each other. For the stick class, there is no problem with unclarity in annotation, however the shape of the stick is often confused with a plant.

5.2 Object detection with MaskRCNN to detect 1 year old and 2 year old branches

5.2.1 Methodology

For this approach only the colour images of the camera on the end-effector and its annotations have been used. There are 2 possible approaches to differentiate between 1 and 2 year old branches: (1) detect objects of class 2 year old branch directly from a deep learning object detector, along with the pixel masks so that the pruning location can be determined; or (2) a 2 stage approach in which candidate masks for 2 year old branches are detected followed by a classification step that decides whether an object is a 1 year or a 2 year old branch. The latter approach can use either a one class (i.e. branch, independent of age) deep learning object detector, or an approach such as Follow the leader which determines the central trunk and accordingly segments the branches without knowing their age. The annotated instance masks were used to train an object detection model (MaskRCNN, Detectron2 implementation). Data augmentation was used to increase the number of 2 year old branches, by flipping them left to right, and top to bottom. This was applied only to the training set. The ResNet101 backbone was used with point rend. Since the annotations also contained the trunk, a 3 class (1 and 2 year old branches and trunk) model was trained. A train/test split ratio of 75/25 was used.

5.2.2 Results

The accuracy on the overall test set (RealSense + ZED) was 0.33, and 0.64 on only the RealSense subset. Figure 25 shows the confusion matrix obtained in terms of the predicted labels, along with an example image with the detections overlaid. Note that the class "spurious" was not defined while training, but consists of all false positives that are not due to any of the other classes.

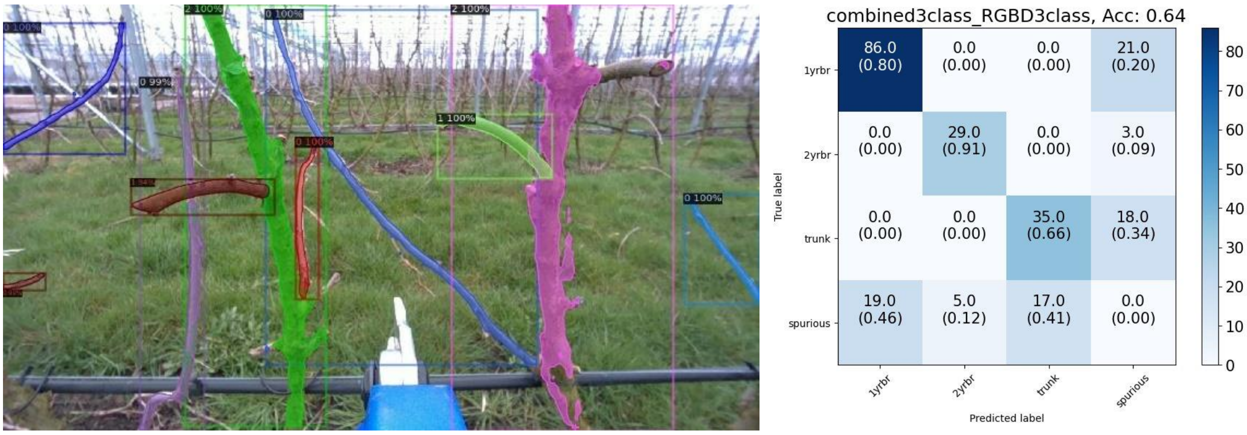

Figure 25 *Results of object detection using MaskRCNN: (left) example image, (right) confusion matrix on RealSense test set.*

By training on specific datasets, i.e. either on the ZED or RealSense datasets only, the test set accuracy is significantly worse as seen in the confusion matrices in Figure 26 which were obtained by training and testing within each of the ZED and RealSense datasets.

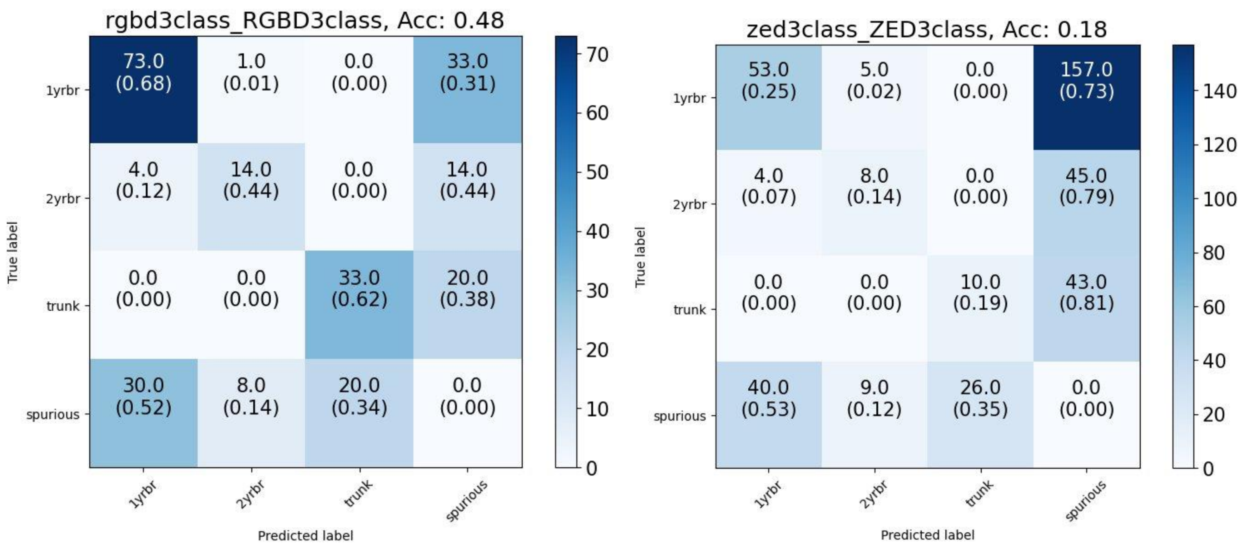

Figure 26 *Confusion matrices obtained using dataset specific training for (left) RealSense dataset, (right) ZED dataset.*

Classification between branches
The performance of deep learning classification was evaluated on both the existing datasets (RealSense and ZED) as listed in Table 1 on page 18.

Classification with Darwin annotated data

Binary (1 or 2 year old branches) or Multi-class (1 year/2 year branches + trunk) classifiers using the 18 and 50 layer ResNet architectures were trained on the masked dataset, with a 75/25 train/test split. The results are summarized in Table 8. Bounding boxes instead of masked branches were also used, although they did not produce an improvement in performance. The best accuracy and Matthew's Correlation Coefficient (MCC) were obtained using ResNet18 for both the 2 and 3 class problems. The confusion matrices are presented in Figure 27.

Table 8 *Accuracy and Matthew's Correlation Coefficient for branch classification.*

Setting	Arch	Accuracy	MCC
3 class, masked	R18	0.83	0.7
	R50	0.83	0.7
3 class, BB	R18	0.81	0.67
	R50	0.82	0.68
2 class, masked	R18	0.84	0.52
	R50	0.82	0.46

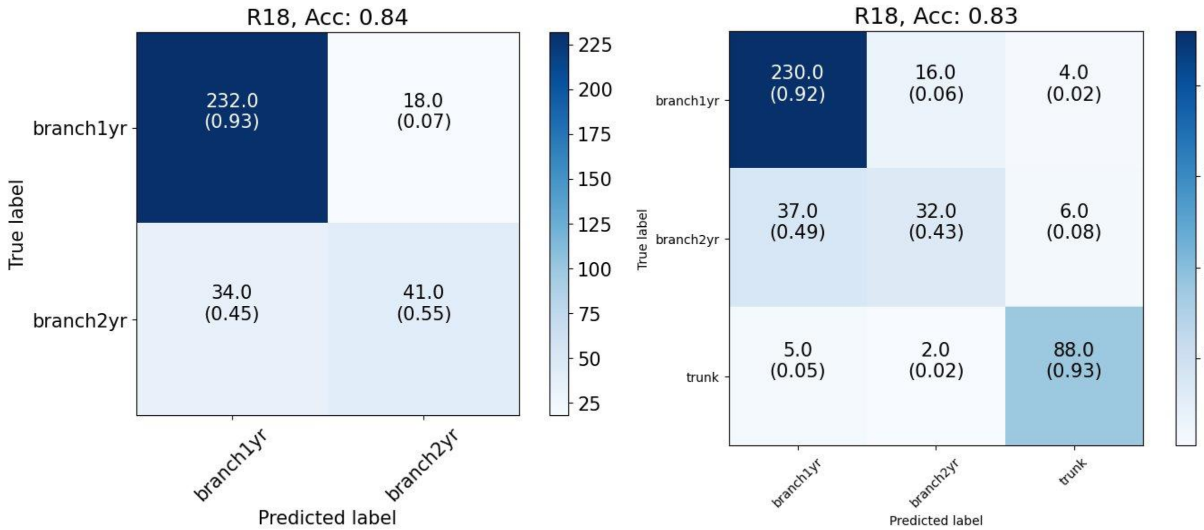

Figure 27 *Confusion matrices obtained using ResNet18 for (left) 2 class case (1 year vs 2 year branches), and (right) 3 class case (1 year and 2 year branches, plus trunks).*

Classification with the output of "Follow the Leader"

Similarly, with the RealSense images from December 2024, a ResNet18 classifier was trained but this time with a reject class consisting of spurious artifacts detected by follow the leader, along with the 1 and 2 year old branches. The classifier obtained an accuracy of 0.76 on the test set, with the confusion matrix shown in Figure 28. The accuracy seems to be dominated by the reject class, which is the largest. However 58% of the 2 year old branches were correctly classified. A further set of images, not part of the initial set, were annotated as a 3-way split. Applying the trained model on this set produced an accuracy of 0.8, again dominated by the reject class. Roughly one in three 2 year old branches were correctly classified.

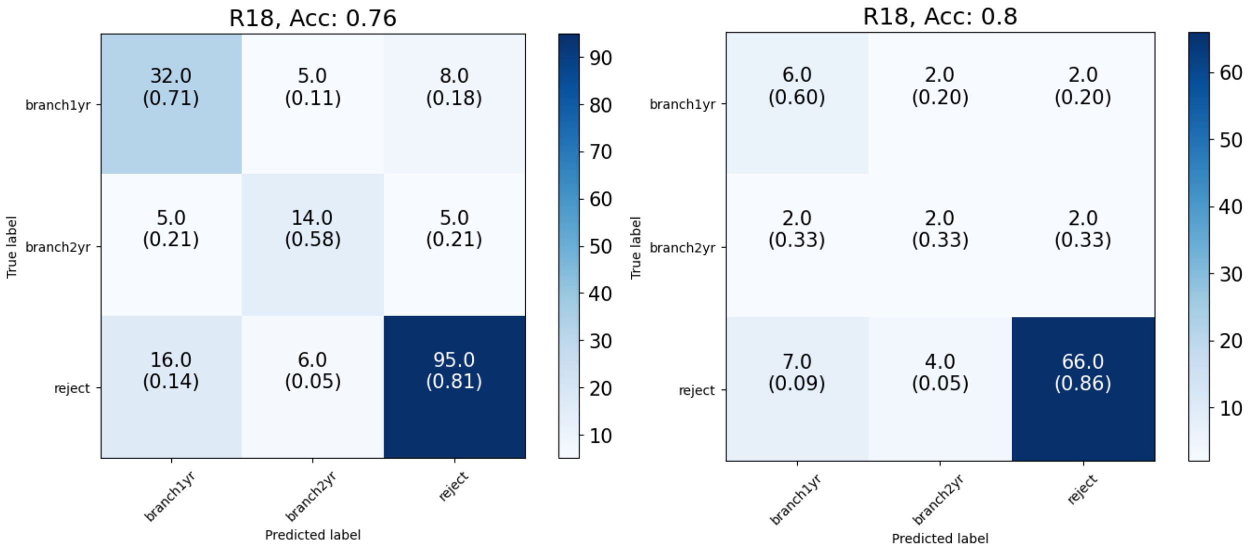

Figure 28 *Confusion matrices obtained using the results of follow the leader. Left matrix show the results obtained with the original test set of images. Right matrix show the results of another test set of images that have not been part of the initial set.*

Computational Requirements

The presented results were obtained on an intel i7 based system with an NVIDIA Titan XP GPU, running Linux Mint 22.0. The object detection using MaskRCNN takes roughly 40 milliseconds per image, whereas the classifier using ResNet has an inference time in the order of 200 milliseconds. The training on the same system took roughly 8 hours for object detection and 3 hours for classification.

6 Discussion and Conclusion

This study provides valuable insights into the potential and challenges of sensor-based pruning automation, laying the groundwork for future development of robots for pruning. Two sensing concepts were investigated for their suitability for robotic pruning of a dormant red currant plant. (1) LiDAR and ToF/colour cameras to scan the plant structure from a distance. (2) a camera system mounted on the robotic end-effector. The different concepts come with a different hardware price tag: The utilized LiDAR and Time of Flight (ToF) camera used in the first concept are relatively expensive. The benefit of the used Leica LiDAR, for instance, is, that it also provides SLAM (Simultaneous Localization and Mapping), making the data immediately useful. Nevertheless, with about €40 000 for this sensor only it will significantly increase the total price of the system. While it is possible to register cameras using a less expensive LiDAR (as demonstrated by the IPI setup), this process is very costly in terms of GPU computing power and requires hours of offline processing. Consequently, this approach will require that for pruning with a robot the scene needs to be visited 2 times: first time for scanning and then for the pruning action. This will require an accurate localization and geo-referencing system on the robot and also poses the risk of not being successful with the pruning action as the scene might have changed, think of moved or broken branches due to wind or other external influences.

Performing SLAM with the ToF camera did not work directly. Therefore, co-registration with the LiDAR would be beneficial to achieve a high-resolution point cloud, although this was not accomplished within the current project. Regarding the instance segmentation with OneFormer3D, we observed that object recognition performed reasonably well, with around 85% of points that are correctly recognized on average per class (see Table 6). However, the standard performance measurement methods are not always the best methods for this use case. For individual plant recognition, when the instance segmentation was joined with semantic segmentation the results were worse than expected. Because of the closeness of the bushes and therefore branches of neighoubourng plants overlapping each other only about 60% of the plants could be correctly identified. In our opinion, these current results are not yet sufficient for controlling a robot. The results suprised us, since we obtained far better performance with data of apple trees where we where under the impression that they might be more difficult. Looking back and during another visit of the Randwijk orchard, we observed that Randwijk has per row plants of different age and sometimes even different training systems. We used one row in training and another row in testing. However, if they are different systems, generalizing might be more difficult than we had anticipated. Another point is that we expected that small bushes/trees perform better than bigger trees, however with the LiDAR we observed that we have also generated less points which in this case has a negative effect on the OneFormer3D results, instead of the positive effect we hoped for. This might be improved in the future with more dense LiDAR point clouds.

The small camera on top of the robotic pruning end-effector used the second concept is much more affordable (less than €300). However, in order to generate a full overview of the plant the camera needs first to be moved with the robot arm along the main leader of the plant. We have shown that the analysis of such images can be done real-time, but the scanning motion is time consuming. And despite the close distance to the plant, the feature extraction from these images turned out to be limited and not robust enough for an autonomous real world application.

For the developed MaskRCNN object detection it can be concluded that the 2 year branches tend to get misclassified as 1 year old ones, more than the other way round. The imbalance in the sizes of the classes also contributes to the skewed classification performance. The present models, while not robust enough for real world usage, can be further optimized and fine-tuned with better quality data. There are also more clear criteria needed for image labelling. Monitoring bushes since the early stages of life or synthetic data could play also a role here. A combination of background removal (for example using depth anything (Yang et al., 2024) and branch segmentation could be a solution for reducing the number of spurious detections. A larger dataset leveraging existing models and unsupervised methods corrected through active learning such as developed by Blok et al. (2022), could help improve the performance.

In conclusion, both sensing concepts investigated will on themselves, at their current stage of development, not be sufficient to deliver the data needed to perform robust and real-time robotic pruning. In the future combining the ToF camera and LiDAR on the platform could provide high-resolution data, addressing the remaining issues with instance segmentation. Additionally, linking the results from both sensor concepts could produce a sufficient dense and detailed 3D environment. With this combined information detailed strategies and motion planning concepts could be developed to control the robotic arm(s) with the pruning tool. An example on how to combine sensors is illustrated in **Figure 29**. In this case, a LiDAR sensor mounted on the base platform can first make a scan of the bush with a slightly higher point density compared to the system researched in this project. After scanning the robot platform would drive to the next tree while keeping track of its relative position. The robotic manipulator system gets information from the base platform sensors about the location and of main and side branches and the camera mounted on the robotic arm does not have to scan the full plant again by scanning top to bottom. Instead the arm can directly go to a side branch position, eventually using the camera on the arm to refine its position and to decide by analysing the images from this camera which branches to prune. The time needed by the robotic arm for pruning would give time to analyse the point cloud data of the next bush collected by the LiDAR. Nevertheless, such a sensor system would require the investment of several €10 000 for the sensors only. It is questionable if this would fit into a viable business case of a future pruning robot. Having a platform that is equipped with such powerful sensing capabilities would offer more possibilities than just pruning. It might be used for other tasks like to follow the development of single plants, whole plantations over time (during one year of even during several years), sense plant health (pest and diseases) or to predict future production.

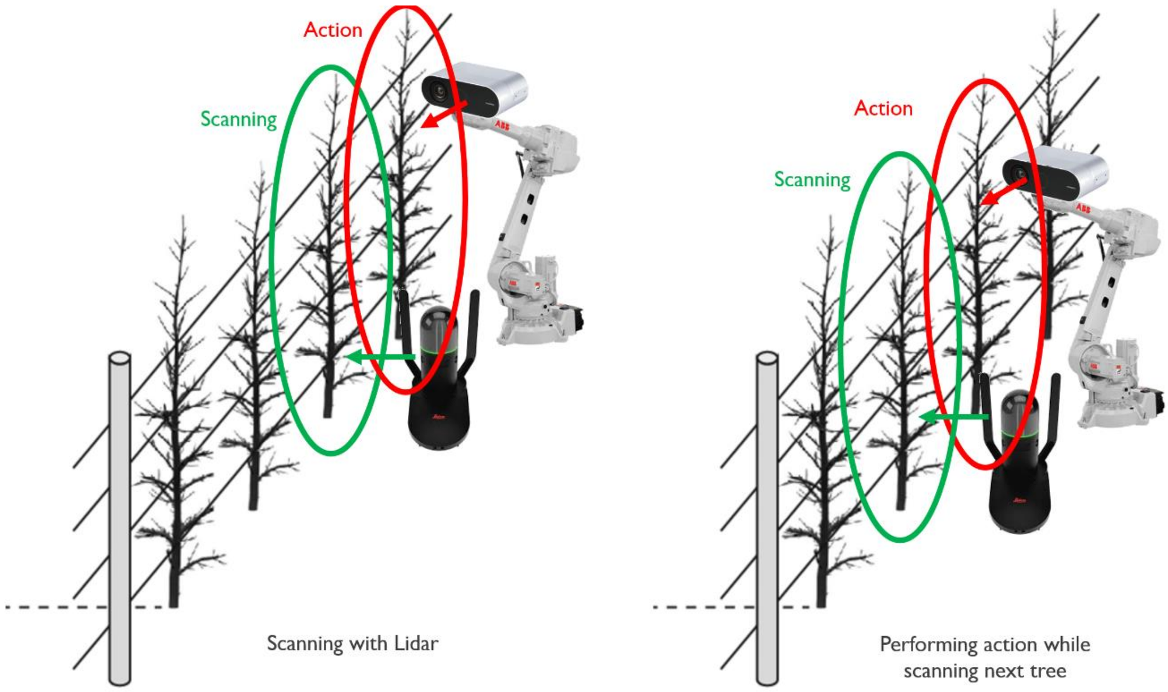

Figure 29 *Concept of future pruning robotic setup by combining sensors on a base platform with a camera on the end-effector.*

The developed sensing algorithms show the high potential of deep learning based approaches to identify and localize different parts of a red currant plant and to identify other objects in the scene, such as poles, ground, supporting wires. Still more work is required to improve the performance of the sensors and algorithms to a level where the outputs can be translated with a sufficient low error rate into robotic control commands for precise pruning execution.

References

Blok, Pieter M., Kootstra, Gert, Elchaoui Elghor, Hakim, Diallo Boubacar, van Evert, Frits K., van Henten, Eldert 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, 2022, 106917, https://doi.org/10.1016/j.compag.2022.106917.

Garrido, D., Rodrigues, R., Augusto Sousa, A., Jacob, J., & Castro Silva, D. (2021). Point Cloud Interaction and Manipulation in Virtual Reality. 2021 5th International Conference on Artificial Intelligence and Virtual Reality (AIVR), 15–20. https://doi.org/10.1145/3480433.3480437.

Kerbl, Bernhard, Georgios Kopanas, Thomas Leimkühler, and George Drettakis (2023). "3d gaussian splatting for real-time radiance field rendering." ACM Trans. Graph. 42, no. 4: 139-1.

Kolodiazhnyi, Maxim, et al. (2024). "Oneformer3d: One transformer for unified point cloud segmentation." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Ravi, Nikhila, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr et al. (2024). Sam 2: Segment anything in images and videos." *arXiv preprint arXiv:2408.00714*.

Vaswani, A. (2017). "Attention is all you need." *Advances in Neural Information Processing Systems*.

Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., & Zhao, H. (2024). Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10371-10381).

You, Alexander, Cindy Grimm, and Joseph R. Davidson (2022). Optical flow-based branch segmentation for complex orchard environments. 2022 IEEE/RSJ International Conference on Intelligent Robots and systems (IROS).

To explore the potential of nature to improve the quality of life

Wageningen University & Research
BU Greenhouse Horticulture
Violierenweg 1, 2665 MV Bleiswijk
P.O. Box 20, 2665 ZG Bleiswijk
The Netherlands
T +31 (0) 317 48 56 06
E glastuinbouw@wur.nl
wur.eu/greenhouse-horticulture

Report WPR-1411

The mission of Wageningen University & Research is "To explore the potential of nature to improve the quality of life". Under the banner Wageningen University & Research, Wageningen University and the specialised research institutes of the Wageningen Research Foundation have joined forces in contributing to finding solutions to important questions in the domain of healthy food and living environment. With its roughly 30 branches, 7,700 employees (7,000 fte), 2,500 PhD and EngD candidates, 13,100 students and over 150,000 participants to WUR's Life Long Learning, Wageningen University & Research is one of the leading organisations in its domain. The unique Wageningen approach lies in its integrated approach to issues and the collaboration between different disciplines.