Ref. Ares(2023)4538653 - 30/06/2023

This project is funded by the European Union
G.A No 101016906

Project Number: 101016906
Start Date of Project: 2021/01/01
Duration: 48 months

Type of document D4.1 – V3.0

Deliverable Agronomic target modelling, detection and localization

Dissemination level	PUBLIC
Submission Date	2023-06-30
Work Package	WP4
Task	T4.1, T4.2
Type	Report
Version	3.0
Author	UPC, UNIROMA1
Approved by	CONSORTIUM

DISCLAIMER:
The sole responsibility for the content of this deliverable rests with the authors. It does not necessarily reflect the opinion of the European Union. Neither the Directorate-General for Communications Networks, Content and Technology, Artificial Intelligence and Digital Industry nor the European Commission are responsible for any use that may be made of the information

This project is funded by the European Union G.A No 101016906

Executive Summary

In this deliverable, we present a real-time method for the detection and localization of the peduncle of table grapes which is robust and can be applied by a robot. The objective is to detect the grape bunch and identify and localize with high precision the point of the peduncle where the robot's cutting tool must reach. The bunch and peduncle detection methods presented in this work rely on a combination of instance segmentation and monocular depth using Convolutional Neural Networks (CNN). The computation of the localization of the grapes and peduncles is done by a combination of different depth techniques.

The methods were tested using a dataset of 1,326 RGB-D images of table grapes that were collected at the Corsira Agricultural Cooperative Society (Aprilia, Italy) in the month of September 2021, using a Realsense D435i camera located at the arm of the CANOPIES two-manipulator robot developed in the project.

In addition, a method to generate new segmentation and detection data is described. This method was designed to be able to reduce covariate shifts and to make continual learning, season after season, easier to achieve.

We also present the detection of pruning points of the vineyard branches by a robot. The objective is to detect the pruning point at the middle of two buds of a 1-year (or 2-year) old branch. However, at the same time the objective is to segment part of the vineyard branches and wire trellis close to the pruning point to avoid robot motion collisions with the environment space when pruning.

The methods were tested using a dataset of 5 video sequences of branches to prune that were collected at the Corsira Agricultural Cooperative Society (Aprilia, Italy) in the month of January 2023, using a Realsense D435i camera located at the arm of the CANOPIES two-manipulator robot.

This project is funded by the European Union G.A No 101016906

Table of Content

1 Introduction 5
2 Robot and sensors 6
3 Detection and localization of grapes and peduncles 7
3.1 System overview 7
3.2 CANOPIES dataset of grape and peduncle images 8
3.3 Instance segmentation in estimated RGB-D images 11
3.4 Experimental results of instance segmentation 12
3.5 Depth estimation of table grapes and peduncles 12
3.6 Experimental results of depth estimation of the peduncles from the robot camera 13
3.7 Weakly and Semi-supervised learning for label generation 14
3.7.1 System overview 14
3.7.2 Pseudo-labels generation experimental results 16
4 Detection and localization of pruning points, branches, and cables 19
4.1 System overview 19
4.2 CANOPIES dataset of pruning images 20
4.3 Detection and localization of the fingertip of the person 21
4.4 Instance segmentation of branches 22
4.5 Detection of wire cables 24
4.6 Experimental results of wire trellis detection 27
4.7 Detection of the pruning points 27
5 Conclusions 31
6 References 32

This project is funded by the European Union G.A No 101016906

Abbreviations and Acronyms

CANOPIES	A Collaborative Paradigm for Human Workers and Multi-Robot Teams in Precision Agriculture Systems
RGB	Red Green Blue
RGB-D	Red Green Blue - Depth
FOV	Field of View
UPC	Universitat Politècnica de Catalunya
IRI	Institut de Robòtica i Informàtica Industrial
UMA	Unidad de Mecanización Agraria
VIA	VGG Image Annotator
Monodepth	Monocular depth
HTML	Hypertext Markup Language
RCNN	Region-based Convolutional Neural Network
CNN	Convolutional Neural Network
IoU	Intersection over Union
AP	Average precision
mAP	Mean average precision
Tvid	Target video sequence
SDet	Source detector
SSeg	Source segmentation network
TDet	Target detector
Tseg	Target segmentation network
WGISD	Wine Grape Instance Segmentation Dataset
TPLG	Detection Pseudo-Labels Generation
SPLG	Segmentation Pseudo-Labels Generation

This project is funded by the European Union G.A No 101016906

1 Introduction

In this deliverable we present the results of Task 4.1 “Agronomic Target Detection and Modelling” and Task 4.2 “Agronomic Target Localization”.

The objective of Task 4.1 is to detect grapes bunches and peduncles that must be harvested by a robot since the robot cannot touch the grapes in order not to damage them. A second objective is to detect the branch point where the robot must prune the branch, and to detect the branches near the pruning point to locate the obstacles and the wire trellis cables to avoid the collision of the robot arm when performing the motion trajectory.

The objective of Task 4.2 is to localize the grapes and the peduncles from the robot's head and wrist, to be used by the robot for harvesting. A second objective is to locate the pruning point and the localization of the branches and cables.

In the following sections, we explain briefly the methods and the tests that were done with datasets of images collected in the agriculture field, in the harvesting period (September), and in the pruning period (January).

The organization of the document is as follows. In section 2, we explain the robot that has been used and the sensors. In section 3, we explain the detection and localization methods for grape bunches and peduncles for harvesting. In section 4, we describe the methods for the detection of the pruning points, the detection of the branches, and finally the detection of the wire trellis. Finally, in section 5, we present the conclusion of the deliverable.

2 Robot and sensors

We have used a robot that is an evolution of the Tiago++ robot manufactured by PAL Robotics. This robot was specifically designed for the CANOPIES European project. It is mounted on the Alitrack platform, a caterpillar platform, and it has two robot arms. Each one has 7 degrees of freedom and in each one of them, it is located a Realsense D435i stereo camera and in the head of the Tiago++ there is another Realsense D435i (see Fig.1.a).

The Realsense D435i camera sensor was configured with a resolution of 1280 × 720 pixels, and frames were captured at a rate of 30 frames per second. The RGB camera has a field of view (FOV) of 69 degrees horizontally and 42 degrees vertically. In terms of depth settings, the camera used stereoscopic depth technology, providing a depth field of view of 87 degrees horizontally and 58 degrees vertically. At maximum resolution, the minimum depth distance (Min-Z) is of 28 cm. The depth output resolution was configured to be 1280 × 720 pixels, the same as the RGB camera, and the depth images were captured at a rate of 30 frames per second. The Realsense D435i mounted on the robot wrist can be seen in Fig.1b.

Figure 1: (a) CANOPIES Tiago++ bi-manipulator; (b) gripper with the Realsense D435i camera on the wrist

3 Detection and localization of grapes and peduncles

3.1 System overview

The CANOPIES project focuses on harvesting table grapes, which is a less developed sector in terms of robotics and automation than other branches of agriculture. Grapes grow on plants called vines or grapevines. Short branches originate from the trunk of the vine, referred to as arms, from which shoots originate. When the green stems or shoots mature and become woody, they are referred to as canes. As can be seen in Fig.2, in the harvest season the grapes have grown in clusters or bunches, which are connected to the cane through the peduncle.

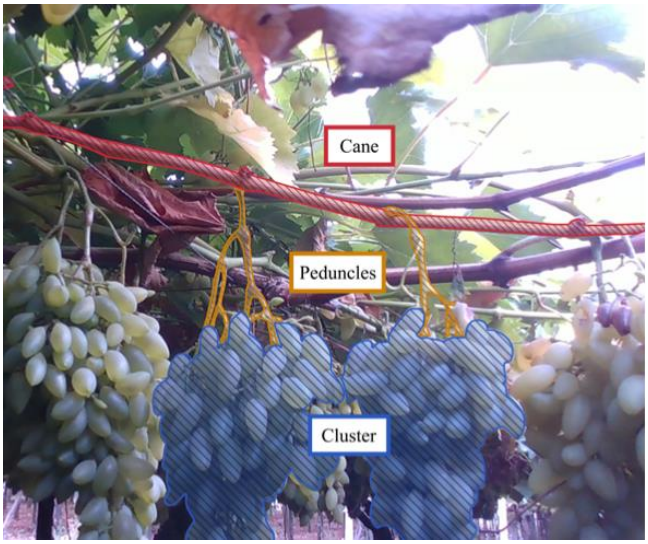

Fig.2: Grapevine parts in the section of the plant closest to the grapes.

The goal is to detect both the grape bunch and the peduncle. To determine the cutting point, it is necessary to have a segmentation map at a pixel level of the peduncle. On the other hand, bunch detection will also be essential when the camera is not close to the vine, especially for selecting the next target to collect. Since the peduncle is of considerably smaller dimensions, it will not be detectable or even visible when observed at a greater distance.

To detect the bunches and their peduncles, we develop a system, which general process flow can be seen in Fig.3.

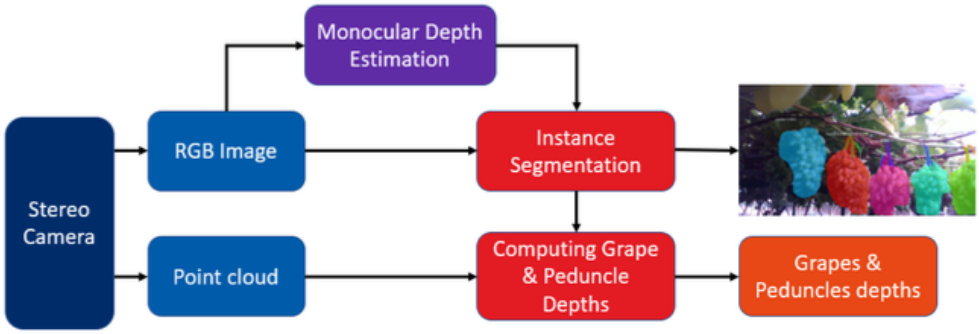

Fig. 3 General process flow of the method

The main blocks of the detection are the following:

- Stereo camera: from the stereo camera is obtained an RGB image and a Point-cloud
- Monocular Depth estimation: from the RGB image is obtained an estimate of the depth map, which is used as the input for the Instance Segmentation module.
- Instance Segmentation: it is the core of the system, which uses as input the monocular depth and the RGB image, and processes them to detect the bunches and peduncles
- Computing Grape & Peduncle Depths: there are several methods that compute the depth of the detected grapes and the localization ((x,y,z) coordinates) of the peduncle point where the robot gripper must go.
- Grapes & Peduncles Depths: the depth methods are fused to obtain the best localization of the harvesting point.

The methods of the main blocks will be explained briefly in the next sections. A deeper explanation of these methods can be found in the article [1] that will be published soon.

3.2 CANOPIES dataset of grape and peduncle images

In September 2021, multiple sequences were recorded in the Corsira agricultural cooperative society vineyard fields by a team of the UPC, La Sapienza and Università Roma Tre researchers of the CANOPIES project. The sequences were recorded using a Realsense D453i camera. In all of them, the RGB images were recorded. Since 3D point cloud estimation is only feasible at a distance greater than 20 cm (in the D400 series manual [2], the working range for the D435i camera is stated to be from 0.2 to 10 meters), the Point-cloud was only recorded in those sequences that did not occur at close distance.

A total of 45 sequences were recorded, with a combined duration of 3,660 seconds (61 minutes). To increase the variability of the data, the location of the camera was changed during the recordings. Images and Point-clouds were captured from outside the vine (point of view of the harvester robot head) as well as from inside (point of view of the end effector of the robot arm). In total, the

sequences contained 76,283 images and 57,547 Point-clouds captured with the Realsense camera. With the use of a second ZED 2 camera, 27,492 images and 22,942 Point-clouds were also recorded. The disk capacity taken up by the entire collection of data was 1 TB. Some part of these data will be open to the public by the end of the project once related research papers have been published

To build a segmentation approach, a part of the dataset was selected for labeling. The sequences in which the peduncles and grape clusters could be seen, totally or partially, were selected, discarding those recorded from too far away to be able to identify the peduncle. Ultimately, 10 sequences were selected from the original dataset, containing 1,326 different images. Some examples can be seen in Fig.4.

Fig.4: Examples from the selected sequences.

To label the images, the open-source tool VGG Image Annotator (VIA) [3] was used. The VIA tool is free to use and allows the annotation of images, videos, and audio. Another advantage is that is a self-contained HTML page that runs offline in a web browser, without requiring any installation. The tool allows different ways to segment images into different regions: rectangle, circle, ellipse, polygonal region or polyline. Since both cluster and peduncle have complex shapes, it was decided to use the polygonal region to segment the shapes (see Fig.5). For each case in each image, two masks were generated: one for the peduncle alone and one containing both the grape cluster and the peduncle.

Fig.5: Grapevine parts in the section of the plant closest to the grapes.

Examples of the ground truth can be seen in Fig. 6.

Fig.6: Ground truth examples from the dataset. Both classes in the dataset are shown: bunches (blue) and peduncles (green).

3.3 Instance segmentation in estimated RGB-D images

Object detection is a combined problem: locating an element in an image and classifying it. The goal is to take an image and obtain a set of bounding boxes framing the objects of interest, each box having a class and a score.

One difficulty to overcome is that the size of an object in an image is not known. Depending on how an image was taken, the scale, orientation, and location of an object will change. Another difficulty that arises is that the number of objects in the image is also variable and unknown. A robust object detection method must be able to overcome these problems. Lighting and occlusion are two extra problems that need to be addressed. Different illumination conditions can cause the same object to look different in distinct pictures. The approach must be robust to these variations. It must also be robust when objects are occluded and not fully visible on camera.

Moreover, the detection of peduncles in a cluttered environment as it is found in vineyard fields, is very complex due to the number of leaves, branches, and partial grape bunches with similar colors and especially, due to the shape and diversity of peduncles that can happen. Using only the RGB image, is not enough because too many ambiguities appear in the detection of peduncles, and for this reason, the use of a depth map can help to eliminate a big part of the background and separate the foreground grape bunches and peduncles from the rest. In this work, we include depth as an essential cue that combined with the RGB image can add the extra information needed to improve image segmentation and peduncle detection. We have used two methods MiDaS [4] and Boosting monodepth method [5] from the RGB image, to obtain the relative Point-cloud and this is used as a new input for the instance segmentation.

We trained a Mask-RCNN network [6] using the RGB and the depth image obtained from the Boosting monodepth method, to obtain a CNN that can detect grape bunches and peduncles better than other known methods.

An example of the detection results can be seen in Fig.7.

Fig.7 Detection of grape bunches and peduncles with their confidence scores

3.4 Experimental results of instance segmentation

We applied this technique on the CANOPIES Grape bunch and peduncle dataset, and the results can be seen in Fig.8. The result shows the comparison of the method using the RGB-D versus RGB,

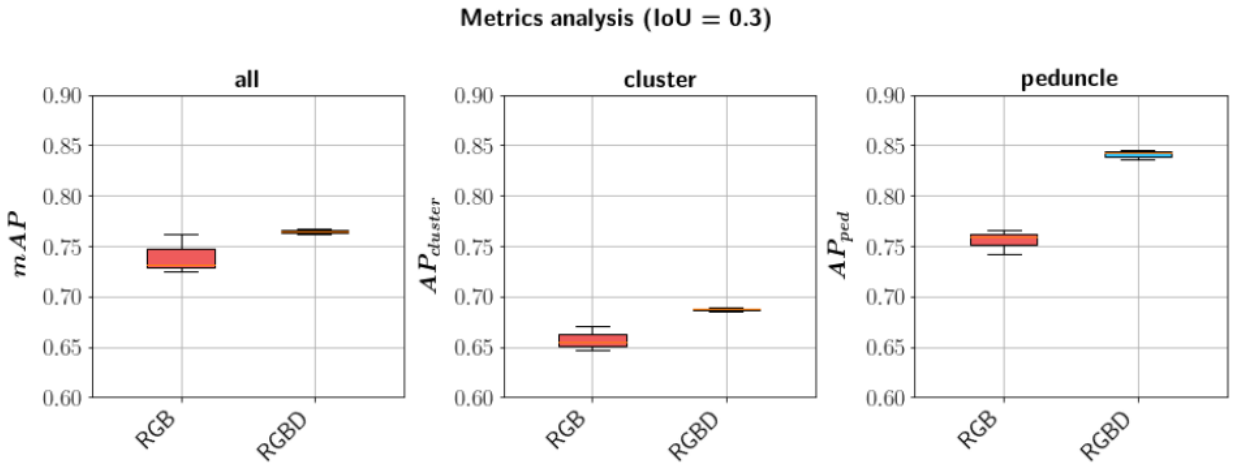

Fig.8 Comparison using RGB versus RGB-D. The graphic shows the mAP metric result for cluster and peduncle detection, the AP metric for cluster detection, and the AP metric for peduncle detection. In all cases the IoU = 0.3.

We have used the common metrics: mean Average Precision (mAP); and Average Precision (AP). The IoU acronym refers to "Intersection over Union". The gain produced between RGB and RGB-D in the CANOPIES Grape Bunch and Peduncle dataset, are clearly shown in the tables 2 and 3 and in Fig.21. The AP metric shows that for cluster detection, for IoU = 0.3, the value is 0.647 for RGB, and 0.688 for RGB-D (6.3% better), and for peduncle detection, for IoU=0.3, the value is 0.802 for RGB and 0.845 for RGB-D (5.4% better). The global difference between RGB and RGB-D is mAP value of 0.725 for RGB and 0.767 for RGB-D (5.8% better). For other IoU values, the results are also better using RGB-D than using RGB images

3.5 Depth estimation of table grapes and peduncles

We have developed four different methods to estimate the depth from the camera to the peduncle. The distance measure to the grape bunch is easier and can be done using only the Point-cloud of the camera sensor, but finding the depth to the peduncle is more complicated. Mainly there are two problems when we use the camera sensor: the Realsense D435 has a minimum distance where it cannot compute the depth of approximately 0.3m; and beyond 1m the camera sensor resolution was not enough to obtain enough peduncle pixels (it has to be taken into account that besides the correspondence problems of the stereo pair, some of the peduncles are very thin, of approximately 1.5mm).

The four techniques that were developed were the following:

- Direct measurement from the Point-cloud
- Peduncle estimation from the cluster
- Dense reconstruction
- Triangulation

The first method, direct measurement from the Point-cloud, requires filtering some of the pixels and making a selection of the points closer to the detection of the peduncle or the grape bunch.

The second method, peduncle estimation from the cluster, requires to do filtering, computing some features of the grape bunch, and obtaining the depth to the closest point and the localization of the peduncle. This method is very good when the peduncle is hard to segment.

The third method, dense reconstruction, requires obtaining a monodepth image from the RGB image, adjusting a polynomial to the results of the monodepth computation, and obtaining the mean value of the predictions. This method works well from 0.4m to 0.9m, although the computational cost is high.

The fourth method, triangulation, obtains the depth using an estimation of the width of the peduncle using the segmented image. This method does not work very well.

Finally, we decide to use the direct measurement and the peduncle estimation from the cluster, although the dense reconstruction method gave very good results, with high computational costs.

3.6 Experimental results of depth estimation of the peduncles from the robot camera

We applied the two depth estimation methods: direct measurement and the peduncle estimation from cluster methods, to the CANOPIES Grape bunch and peduncle dataset, and the results can be seen in Fig.9.

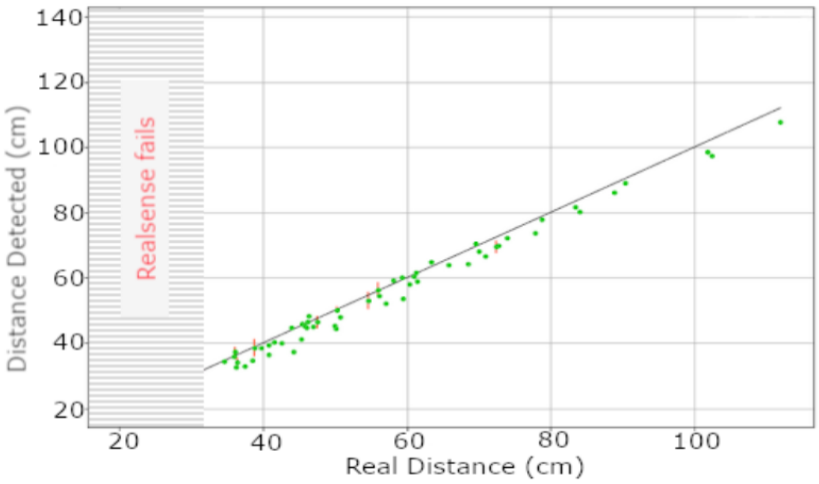

Fig.9 Comparison error between the ground truth distance measured with a laser (horizontal axis) and the estimated distance (vertical axis), using the direct measurement and the peduncle estimation from cluster methods

The mean average depth estimation error was 2.66cm, which is good enough for robot harvesting operations.

3.7 Weakly and Semi-supervised learning for label generation

As mentioned before, the availability of labeled data is one of the critical aspects of this project when thinking of data-driven approaches for perception. Since this is also a common challenge in Precision Agriculture in general, during the initial activities of task T4.1, a system to better leverage the small available quantity of labeled data from public datasets was designed. The proposed system uses this initial data to pseudo-label the data collected by the consortium partners on the Aprilia filed, to have some data that helps in reducing the covariate shift. This pseudo-labelled data, together with the hand labeled one, can lead to better performances of the detection and segmentation algorithms. In this section, an overview of the system is given, while the details can be found in the related scientific publications [8][9].

3.7.1 System overview

The architecture of the system is depicted in Figure 10.

This project is funded by the European Union G.A No 101016906

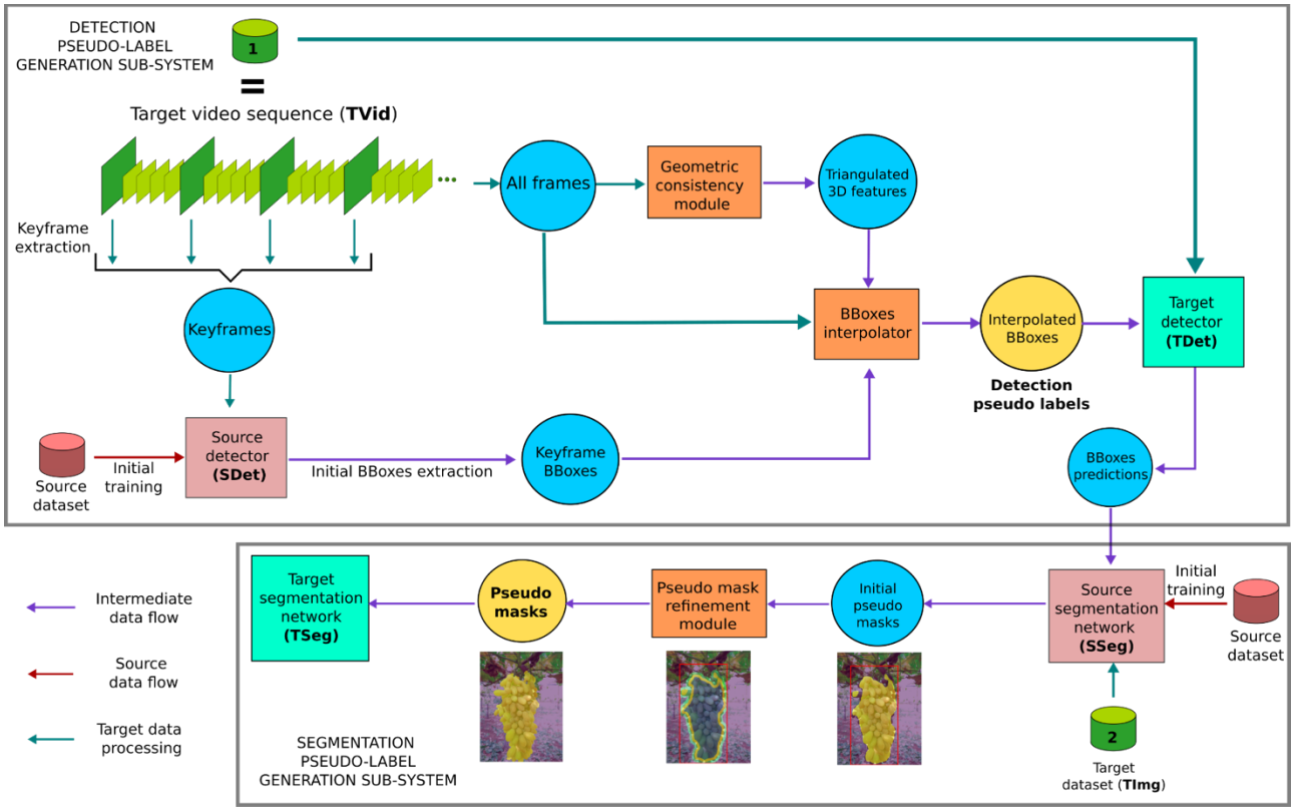

Fig.10: This figure shows the complete system architecture. The inputs are a source dataset (red cylinder) and a video collected on the field by the robot or a farmer (first green cylinder, TVid, expanded as a sequence of frames to show the keyframe selection process). SDet and SSeg (light red blocks) are the initial detection and segmentation networks trained only on the source dataset. All the intermediate computing blocks are depicted in orange, while the intermediate outputs are in blue circles. Both the pseudo bounding boxes and pseudo masks produced are depicted in yellow, while the detection and segmentation networks trained on these new labels (TDet and TSeg) are depicted in light green. The data flow in the system is also color coded as per legend.

The main inspiring principle of this system is the economy of data labelling and data reuse. For this reason, the only two sources of data are the source dataset (data available from a similar task) and a video collected on the target field. The source dataset is used to train the initial detection and segmentation models, namely the Source Detector Network (SDet) and the Source Segmentation Network (SSeg). SDet is not perfectly tuned on the target environment, but it can still be used on selected frames of the video input that we call keyframes. To keep this solution simple, we consider equally spaced keyframes starting from the first one, but other strategies could be devised. A set of initial bounding boxes is extracted from this keyframe, using a high confidence threshold, to limit the false positives. Then, the whole video is passed in a Geometric Consistency block (GC block) that extracts features from each frame and associates them. This check is based on Structure-from-motion algorithms to estimate bounding box motion. Using this geometric information, together with the initial bounding boxes extracted from the keyframes, it is possible to interpolate the bounding boxes positions for the remaining frames with high accuracy. These new bounding boxes are our

pseudo-labels for training the detector on the target environment, which we call Target Detector (TDet).

The Detection Pseudo-Labels Generation (DPLG) sub-system could be used independently by the Segmentation Pseudo-Labels Generation (SPLG). To prove the effectiveness of the approach, we compare the performance of TDet on the bunches tracking problem, i.e., counting the number of grape bunches by counting the instances tracked along a video. This problem is relevant since it can be used for yield estimation purposes. We tested two different tracking algorithms, and the details can be found in the publications.

The goal of the second part of the system is to generate pseudo masks for training an instance segmentation network. This sub-system can be seen both as an independent pseudo label generator, or as part of a bigger system such as the one we describe here. As mentioned before, the SSeg is trained only on source data and is not able to produce good segmentation masks on the Target Data. However, it is possible to give the network some information cues that can greatly improve the mask estimates. The first one is the bounding box region in which the instance should be segmented. This cue comes easily from the previous step of pseudo bounding box generation, but it could be produced otherwise. This generates the initial pseudo masks. It is possible to use these pseudo masks for training the Target Segmentation Network (TSeg) but this would lead to poor performances due to confirmation bias. We need therefore to inject external information from other cues that we have. In our system, this is the role of the pseudo masks refining block. Three different solutions for refinement have been tested, and from the tests one method was chosen as the best one. Thanks to these refined pseudo masks it is finally possible to train the TSeg Network. In the following section the results of the refining strategies are reported and a comparison the performance of TSeg with SSeg is given.

3.7.2 Pseudo-labels generation experimental results

3.7.2.1 Metrics

The metrics used for this experiment come from the MS-COCO dataset [11], as per common practice. The main figures for both detection and segmentation are Precision, Recall, Average Precision (AP), and mean Average Precision (mAP) computed at different thresholds of overlapping (measured by Intersection-over-Union score), in particular, we consider mAP@0.5 and mAP@0.75.

3.7.2.2 Results and discussion

In all the experiments, the source data comes from the “Wine Grape Instance Segmentation Dataset” (WGISD) [10]. Since this data comes from wine grapes and the image are collected in a different environment and illumination condition than the target data, this is perfect to show the functioning of the procedure. Examples of these images are given in Fig.11. For more details on the target data used for the following experiments, please refer to [8][9].

Training the SDet and SSeg with WGISD data and testing it directly on Target data gives the results shown in Table 1 and 2.

Dataset	*Precision*	*Recall*	$mAP_{0.5}$	$mAP_{0.5:0.95}$
WGISD	89.10	83.18	89.68	62.50
T_{img}	83.20	55.60	68.30	42.70

Table 1: Results obtained by the detector trained using only the source dataset, on WGISD dataset and static images collected on the Aprilia field (Timg).

Dataset	Task	AP	AP_{50}	AP_{75}
WGISD	Detection	53.40	87.02	57.36
	Segmentation	53.60	89.44	55.41
TSeg	Detection	32.65	60.40	30.37
	Segmentation	32.88	65.40	34.77

Table 2: Evaluation of an SSeg (Mask RCNN) model on the masked test set of the WGISD (27 images) and TSeg dataset (20 images) using some of the COCO metrics.

Fig11: Left: example of target data; Right: example of WGISD data (source dataset)

The two tables show how the difference in performance due to covariate shift for detection and segmentation is 20.38% and 24.04% mAP@0.5, respectively.

After producing pseudo-labels using the proposed system, and retraining the detection and segmentation algorithm on both WGISD and pseudo-labelled target data, the performance has a considerable improvement, as shown in Tables 3 and 4.

Model	*Precision*	*Recall*	$mAP@0.5$	$mAP@0.5:0.95$
SDet	0.62	0.59	0.55	0.21
TDet	0.74	0.60	0.65	0.23

Table 3: Comparison of the detector trained only with source data (SDet) and with the pseudo labels generated from the videos (TDet) and tested on the target data (TVid).

Model	$mAP_{0.5:0.95}$	$mAP_{0.5}$	$mAP_{0.75}$
Baseline	32.88	65.40	34.77
TSeg (w/o Refinement)	48.43	83.06	53.12
TSeg	49.56	81.03	57.70

Table 4: Comparison of the results obtained by TSeg (Mask R-CNN) on the TImg test set, trained with different datasets and pseudo-mask processing methods.

The results shown in those Tables are obtained by just using one video sequence collected from the target field and pseudo-labeled with the proposed system. Adding more sequences can lead to better performances. In addition, the system is general enough to be applied to other crop types, or different domains.

4 Detection and localization of pruning points, branches, and cables

4.1 System overview

Pruning table grape vineyards is a critical aspect of grapevine management that involves the deliberate removal of vine portions to achieve optimal growth, fruit quality, and yield. Pruning is required to keep grapevines healthy and productive, and it needs careful attention to the timing, technique, and frequency of pruning cuts. Pruning properly helps to control vine size and shape, improves air circulation and sunshine penetration, and encourages the development of strong, disease-resistant vines that yield high-quality grapes.

The time for the pruning operation in table grape vineyards is determined by several factors, including the size of the vineyard, the number of employees available, the training system employed, and the pruning technique used. Pruning is usually done during the vine's dormant season, which is typically from late fall to early spring when no leaves are present. In general, pruning a single vine can take anything from a few minutes to several hours, depending on the complexity of the vine and the skill level of the worker: a large commercial vineyard might require a team of workers for several weeks to complete.

The pruning operation is a complex one that requires analyzing all the trees and looking at which branches must be removed to find the best way that the tree to grow in the next year. Usually, the pruning operation requires having a complete view of the tree and selecting the pruning points of the selected branches, for example of the first, second, or third year. In general, efficient pruning procedures rely on expert pruners' experience, and the decision-making process is frequently structured in the form of a set of guidelines that vary depending on the status of each plant.

More specifically, applying these criteria necessitates rigorous identification of branch kinds and plant structure, as well as detection of nodes and their location and orientation on specific branches. Therefore, automatizing vine pruning is especially hard for a robot (and even humans).

Because of the complexity of the pruning process, in this task, we have simplified the pruning process after talking with the company Agrimessina (a partner of the CANOPIES consortium that is a producer of high-quality table grapes in Italy) and UMA, the "Unidad de Mecanización Agraria" of the UPC.

The new operations look for the collaboration between the human and the robot to reduce the pruning tasks in the following ones:

- First, a farmer operator visits the grape field together with the CANOPIES robot.
- The farmer indicates using the fingertip, the pruning points and the robot captures the image where appears the farmer's fingertip indicating the pruning point (see Fig.12).
- The robot captures the image and the localization of the image in the robot frame reference
- The image is processed, obtaining the fingertip point, the orientation and localization of the fingertip and the branches and wire cables that are around the pruning point.
- All this information is then sent to the robot to do the pruning process.

Fig.12 A person indicating the pruning point to the robot using his fingertip. The image has been taken from the wrist camera of the robot

In this deliverable D4.1 of the Agronomic Oriented-Perception work package, we will only describe the process of capturing and processing the image, to locate the pointed point in a branch, the branches closed to the pruning point and the wire cables seen in the processed image. The rest of the operations like how to perform the cut or putting the bunch in the box, will be described in other deliverables.

In this section, we will start describing the CANOPIES dataset of pruning images, the instance segmentation method of branches, the detection of buds, the detection of wire cables, and the results obtained.

4.2 CANOPIES dataset of pruning images

The data collection was done in January / February 2023 in the Corsira agricultural cooperative society vineyard fields by a team of the UPC and La Sapienza researchers of the CANOPIES project. The camera setup was the same as it is described in section 2.

The sequences were recorded using a Realsense D435i camera. In all of them, the RGB images and the Point-cloud were recorded. A total of 7 video sequences were recorded. To increase the variability of the data, the location of the camera was changed during the recordings.

A robot wrist camera view can be seen in Fig.13, and an image view captured from the camera wrist sensor can be seen in Fig.14.

All the captured images are positioned with respect to the camera framework.

Fig.13 Two views of the branches to be pruned. The left one is the image captured by the camera located in the robot wrist. The right one is the robot extended to prune the branch.

Fig.14 An image of the branches and the wire cables seen from the camera wrist sensor

4.3 Detection and localization of the fingertip of the person

In this work, we use Mediapipe real-time hand tracking [7] to detect the fingertip of the human in the vineyard environment. The fingertip indicates the approximate location of the pruning point,

Fig.12, and the robot will then proceed to identify the cutting location in each 3D sphere containing the branch to prune.

We have developed a software package to obtain the precise location of the fingertip and compute its position with respect to the reference framework of the camera and then of the robot. In this way, we can obtain the fingertip position with respect to the robot's reference framework.

4.4 Instance segmentation of branches

The objective of this part is to provide a depth map of the plant for the Point-cloud reconstruction section which will then take care of the 3D model of the vineyard tree.

The raw Point-cloud information published directly from the Realsense resulted sloppy and un-reliable for the project's purposes and the depth image contained the necessary information about the vine, but with a scarce level of detail Fig.15. On the other hand, in the RGB image, the plant was clearly visible due to the high contrast with the background sky. The idea is then to segment the plant in the RGB image and overlap it with the depth information to remove the background noise and isolate the branches of interest.

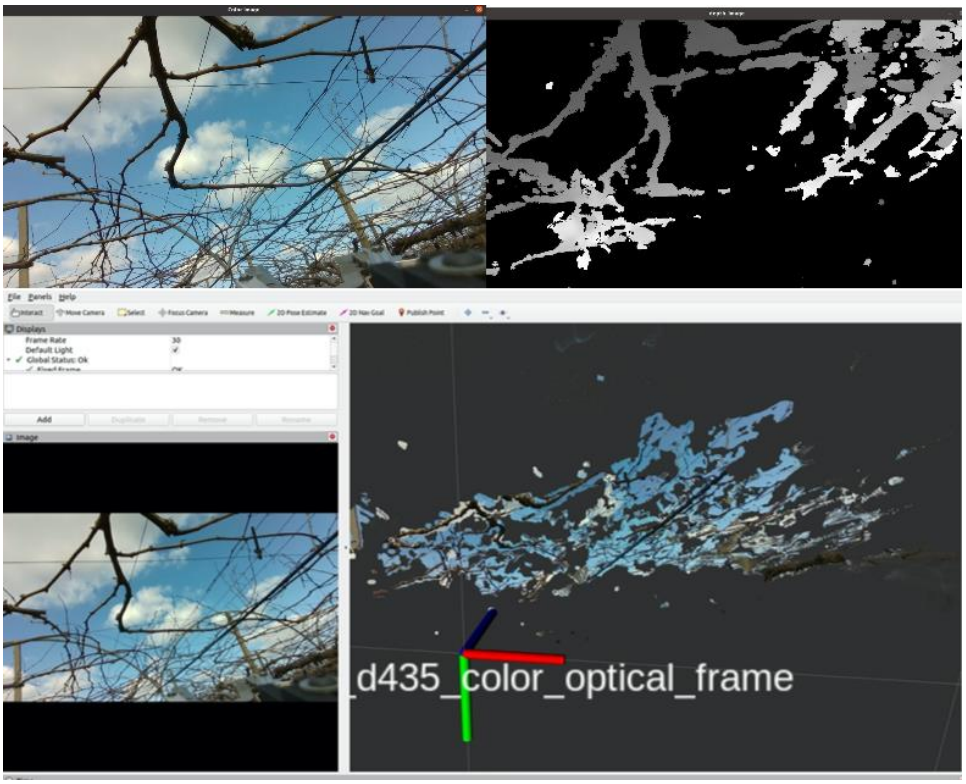

Fig.15 Realsense raw Color image, raw Depth image, raw Point Cloud

Then the segmentation was performed using different techniques to separate the branches from the background and also filter the resulting image using the depth information. The results using segmentation can be seen in Fig.16 and after the filtering step, the results can be seen in Fig.17.

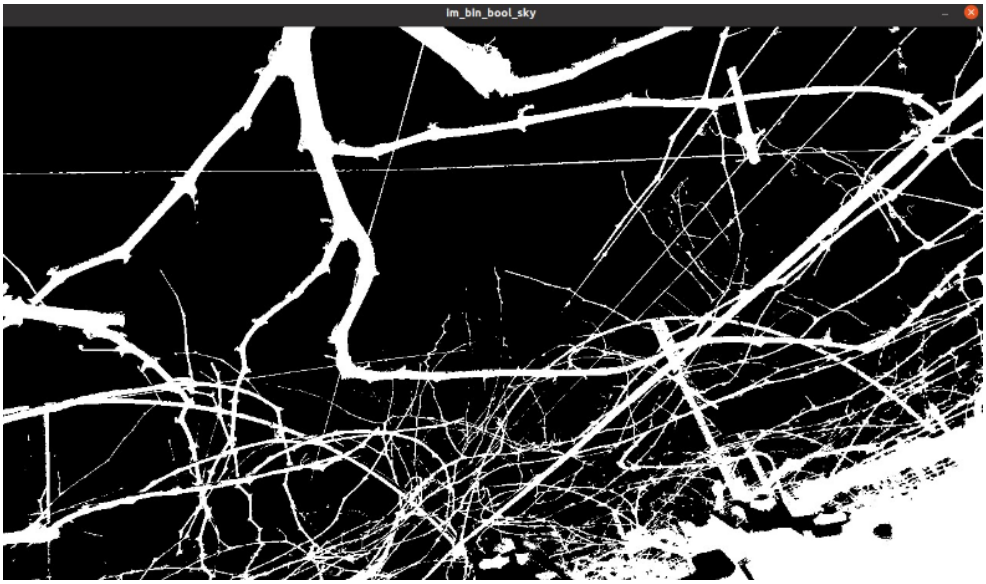

Fig. 16 First segmentation process

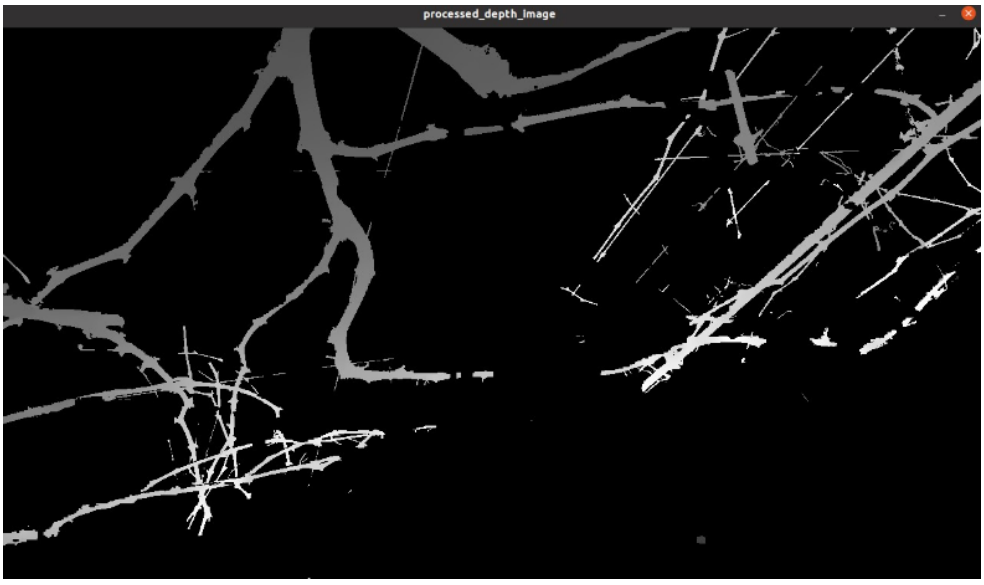

Fig.17 Depth information fused with first segmentation process

Then, we developed some techniques to clean and reconstruct the branches and finally separate them from other obstacles as the human arm and face. Fig. 18 shows the clustering of what is a branch (blur color) and a part of a human (orange and white color).

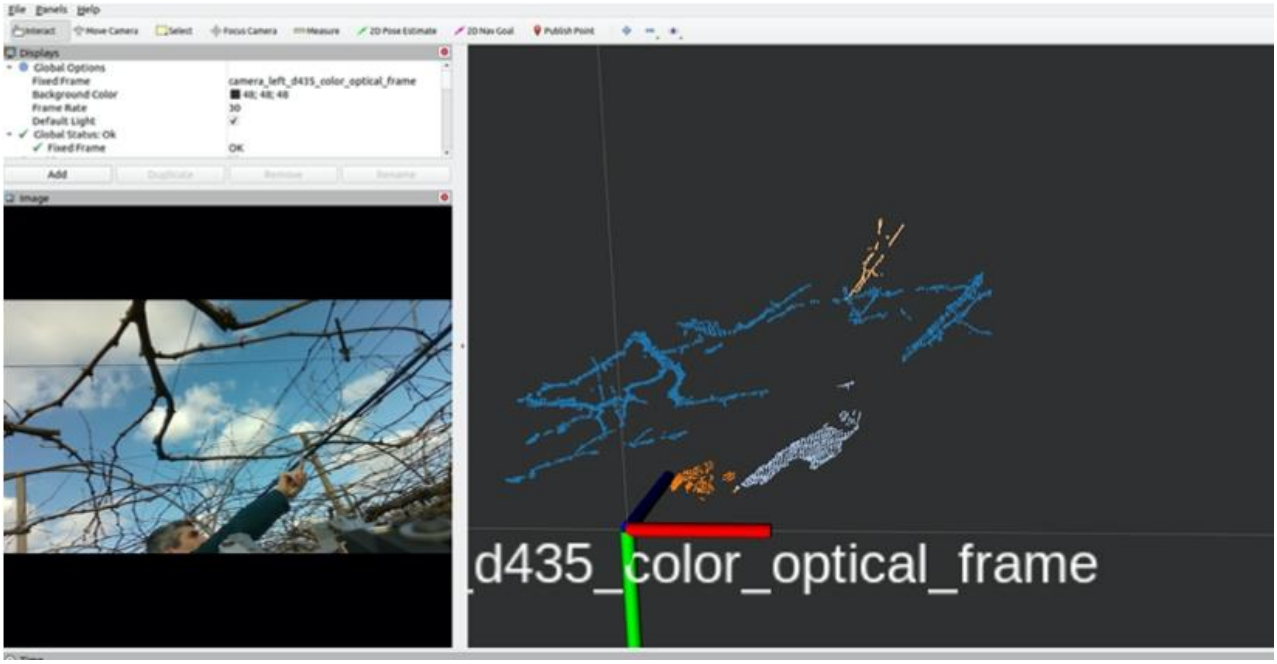

Fig.18 Cluster separation of the branches (blue color) from the human (orange and white colors)

4.5 Detection of wire cables

In a double-roof structure for table grape vineyards, cables are placed horizontally and vertically at a height of two meters from the ground. This displacement encourages the vines to grow radially from the trunk and intertwine with each other, forming a roof of first-year branches that will produce grapes for the harvesting season. During the pruning seasons, no grapes or leaves are present, and the roof is presented as a complicated entanglement between the youngest branches, coming from multiple vines, and the wire structure.

For robotic precision agriculture, identifying and recognizing the cables is crucial for the pruning procedure. In fact, they not only represent an obstacle for the manipulator, but also indicate 3D lines along which it is prohibited to cut. If a pruning point lies next to a cable, which is a common occurrence because the branches of the first and second years lay on the wired roof, the worker must separate the branch from the wire before cutting. This task results extremely difficult for robots to accomplish, requiring the use of two manipulators or a suited gripper.

Identifying the cables is not a simple task. Their diameter is approximately 2mm, which means that with a camera situated beneath the roof, information about the presence of cables is obtained from

only a few pixels in the image. Their depth is rarely measured by the camera making it impossible to directly know their distance with direct measuring.

To do this detection, we divided the problem into three parts:

- Wires identification and depth estimation
- Wires plane identification
- Wires plane projection

Wires identification is the process to detect the wire in the RGB image and estimate their depth. We used some filtering process to separate the wires from the background, then we binarized and thinned the image. The results can be seen in Fig.19.

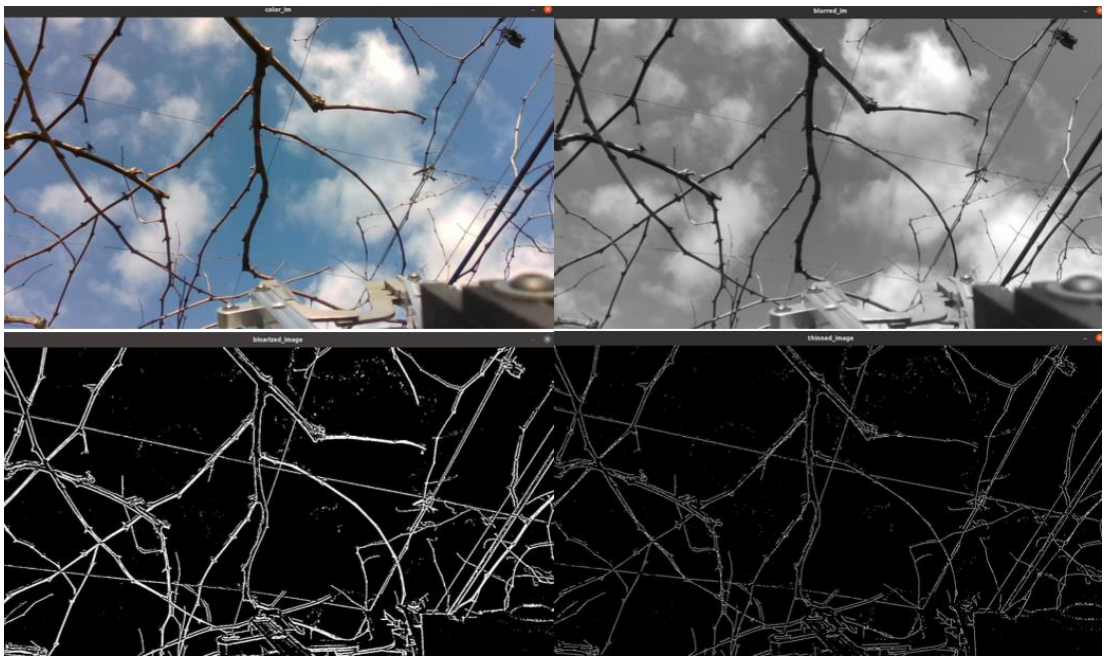

Fig.19 Color, gray level, binarization, and thinned image

Finally, we looked for straight lines as can be seen in Fig.20.

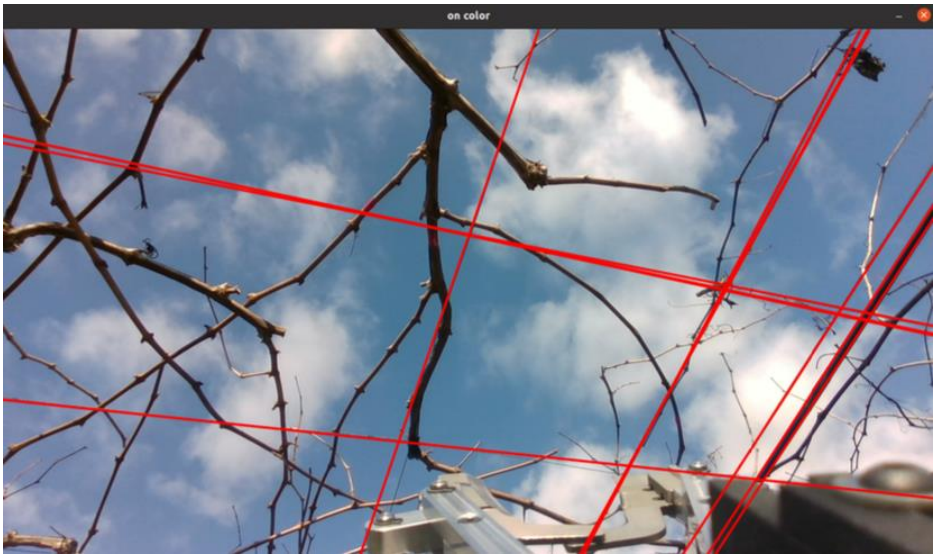

Fig.20 Straight line results

Then, we look for the wire plane generation and compute the wire plane projection. The reason for doing these operations is to eliminate false positive and negative wire cables. This process involves computing the wire plane using the previous segmented information and then obtaining the 3D plane expressed in the camera coordinates. The result of these two processes can be seen in Fig.21.

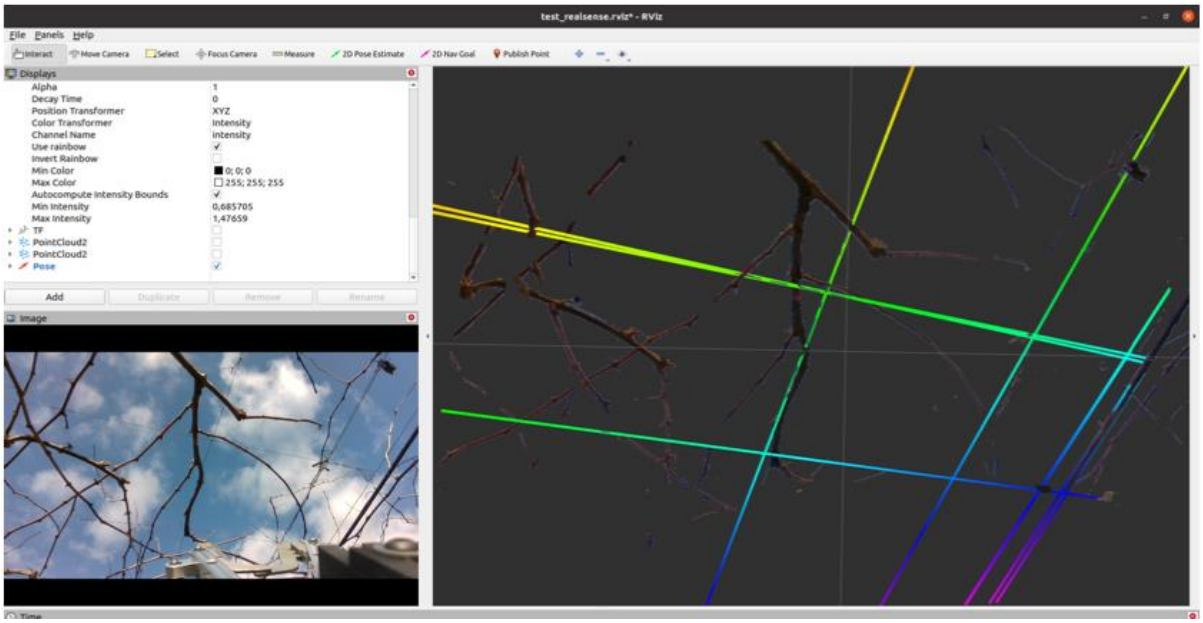

Fig.21 Wires reconstruction final result

This project is funded by the European Union G.A No 101016906

4.6 Experimental results of wire trellis detection

To create a ground truth of the wire cables, to be used for comparison purposes, we manually labelled the wire cables in all the images.

Then, we applied these methods in the CANOPIES dataset of pruning images and the results were the following:

Results	
Number of measurements	700
Number of lines to detect	4000
Numbers of detected lines	3380
Number of missed lines	620
Number of false positives	276
Percentage of detected lines	0.845
Percentage of missed lines	0.155

Fig.22 Wire trellis detection results

The method shows an 84,5% detection rate with 7% of false positives.

4.7 Detection of the pruning points

As we have mentioned before, the CANOPIES process of pruning point detection has been reduced to analyze the buds that are in a sphere centered in the fingertip point of the human operator and then compute the localization of the pruning point. To detect the pruning point, we must detect the extreme buds in the same tree branch and look for the middle point.

The work is divided into three tasks:

- Branch detection: The branch is segmented from the RGB image and its depth information is retrieved.
- Bud detection: The system looks for the buds that are in the branch and inside of the sphere
- Pruning point identification: The placing of the pruning point is given as output.

Branch detection

The RGB and depth images from the camera are received first, after which the pose is extracted from the database and decomposed into each component. The index finger's (x, y, z) coordinates are converted into (u, v) pixel coordinates, then a squared section of the image having as center the (u,v) pair is sliced out. The segmentation explained in Section 4.4 is applied removing the background sky and leaving the cropped image with only cables and branches. The result of these processes can be seen in Fig.23.

Fig. 23 Image with pruning area and cropped image processes

Bud detection

In table grape vines, first-year branches are generally larger than in wine grape vine branches. In fact, the buds protrude from the branch for at least half the diameter, creating recognizable " spikes" that can be exploited for the recognition of the buds themselves. The general idea is to produce a skeletonization of the branch in which each node matches the presence of a bud.

The branch mask discovered in the preceding stage needs to be improved as a first step. Smaller "spikes" along the trunk of the branch can be produced by the watershed segmentation easily absorbing small bits of cable, and occasionally flaws from the lightness segmentation can also provide this effect. It is created the first skeletonization of the branch mask. The image's non-zero pixels are turned into an array and then provided to numpy, which produces the first-degree polynomial of the points that best fit the data. The branch mask's general rotation is then determined by extracting the line's slope. Several morphological operations are used and the results can be seen in Fig.24.

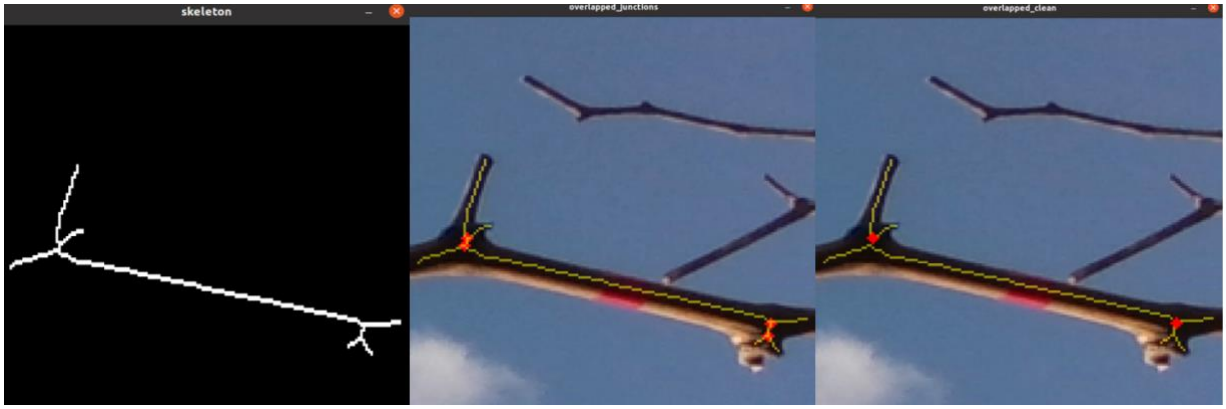

Fig.24 Skeleton, skeleton junctions, and selected buds

Pruning point identification

As we have commented in section 4.1, system overview, to reduce the complexity of the selection of the pruning points, the farmer indicates using the fingertip, the pruning points and the robot captures the image where appears the farmer's fingertip indicating the pruning point (see Fig.12).

The pose estimation of the pruning point must prevent selecting a point too close to the buds because it can easily harm the plant. For this, the pruning point must be positioned in the midst of two buds and oriented so that the cutting tool gets positioned perpendicularly to the branch.

If only one bud is found or no buds are detected, the method returns the pose taken from the database as the closest guess. If two or more buds are discovered, the pruning point is calculated using the two buds closest to the center of the image.

The pixels are then averaged to find a midpoint.

By calculating the distance between the mean pixel and all the pixels in the skeleton, the cutting point is relocated to the closest skeleton pixel identified to ensure that the position is as close to the center of the branch's diameter as possible. Fig.25 shows the cutting zone, identified buds, and the founded pruning point.

This project is funded by the European Union
G.A No 101016906

Fig.25 Pruning point in the whole image

5 Conclusions

In this deliverable, we have described the methods to detect the grape bunches and peduncles and the pruning points to cut the branches.

The grape branches and peduncles are detected using a convolutional neural net that can classify the grape bunches and the peduncles using an RGB image and a monodepth image as inputs. Moreover, a new segmentation and detection method has been proposed, which has the characteristic to reduce the covariate shifts and use continual learning.

The precise localization of the grape bunches and peduncles is done by applying several methods that estimate the localization of the peduncles when they are not seen.

The pruning process has been simplified to prune the vineyard branches without the need to have all the vineyard tree information. It is a collaborative work, between a human operator that first indicates the pruning points to the robot, and then the robot looks for the cutting point. The method combines several methods of instance segmentation and identification of the buds that are close to the pruning point.

It has finally developed, a method to detect the wire cables and the vineyard branches in order that the robot can do the path trajectory avoiding the obstacles.

6 References

[1] Gabriel Coll-Ribes, Iván J. Torres-Rodríguez, Antoni Grau, Edmundo Guerra, Alberto Sanfeliu (2023) Real-time detection and depth estimation of table grapes and peduncles for robot harvesting, combining monocular depth estimation and CNN methods. Submitted (it has already passed the first filter and we are waiting for the final decision).

[2] Intel. Intel Realsense D400 Series Product Family.

[3] A. Dutta, A. Gupta, and A. Zissermann. VGG image annotator (VIA).

http://www.robots.ox.ac.uk/ vgg/software/via/, 2016.

[4] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, V. Koltun, Towards robust monoc- ular depth estimation: Mixing datasets for zero-shot cross-dataset transfer, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (3) (2022)

[5] S. M. H. Miangoleh, S. Dille, L. Mai, S. Paris, Y. Aksoy, Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 9685–9694.

[6] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[7] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020.

[8] Ciarfuglia, Thomas A., et al. "Pseudo-label Generation for Agricultural Robotics Applications." 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, 2022.

[9] Ciarfuglia, Thomas A., et al. "Weakly and semi-supervised detection, segmentation and tracking of table grapes with limited and noisy data." Computers and Electronics in Agriculture 205 (2023): 107624.

[10] Thiago T. Santos, Leonardo L. de Souza, Andreza A. dos Santos, and Sandra Avila. Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. Computers and Electronics in Agriculture, 170:105247, 2020.

[11] Tsung-Yi Lin, Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., … Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. CoRR, abs/1405.0312. Retrieved from http://arxiv.org/abs/1405.0312