POLITECNICO DI TORINO
Repository ISTITUZIONALE

GPS-free autonomous navigation in cluttered tree rows with deep semantic segmentation

*Original*
GPS-free autonomous navigation in cluttered tree rows with deep semantic segmentation / Navone, A., Martini, M., Ambrosio, M., Ostuni, A., Angarano, S., Chiaberge, M.. - In: ROBOTICS AND AUTONOMOUS SYSTEMS. - ISSN 0921-8890. - ELETTRONICO. - 183:(2025). [10.1016/j.robot.2024.104854]

*Availability:*
This version is available at: 11583/2994384 since: 2024-11-14T10:05:12Z

*Publisher:*
Elsevier

*Published*
DOI:10.1016/j.robot.2024.104854

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

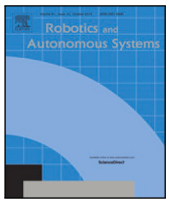
11 November 2026

Robotics and Autonomous Systems 183 (2025) 104854

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot



# GPS-free autonomous navigation in cluttered tree rows with deep semantic segmentation☆,☆☆

Alessandro Navone *, Mauro Martini, Marco Ambrosio, Andrea Ostuni, Simone Angarano, Marcello Chiaberge

*Department of Electronics and Telecommunications, Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino, 10129, TO, Italy*
*PIC4SeR, Politecnico di Torino Interdepartmental Center for Service Robotics, Corso Ferrucci 112, Torino, 10141, TO, Italy*

ARTICLE INFO

*Keywords:*
Autonomous navigation
Service robotics
Semantic segmentation
Precision agriculture

ABSTRACT

Segmentation-based autonomous navigation has recently been presented as an appealing approach to guiding robotic platforms through crop rows without requiring perfect GPS localization. Nevertheless, current techniques are restricted to situations where the distinct separation between the plants and the sky allows for the identification of the row's center. However, tall, dense vegetation, such as high tree rows and orchards, is the primary cause of GPS signal blockage. In this study, we increase the overall robustness and adaptability of the control algorithm by extending the segmentation-based robotic guiding to those cases where canopies and branches occlude the sky and prevent the utilization of GPS and earlier approaches. An efficient Deep Neural Network architecture has been used to address semantic segmentation, performing the training with synthetic data only. Numerous vineyards and tree fields have undergone extensive testing in both simulation and real world to show the solution's competitive benefits. The system achieved unseen results in orchards, with a Mean Average Error smaller than 9% of the maximum width of each row, improving state-of-the-art algorithms by disclosing new scenarios such as close canopy crops. The official code can be found at: https://github.com/PIC4SeR/SegMinNavigation.git.

## 1. Introduction

Precision agriculture has been pushing technological limits to maximize crop yield, boost agricultural operations efficiency, and minimize waste [1]. Contemporary agricultural systems need to be able to gather synthetic essential information from the environment, make or recommend the best decisions based on that knowledge, and carry out those decisions with extreme speed and precision. Deep learning techniques have demonstrated considerable potential in creating these systems by evaluating data from numerous sources, enabling large-scale, high-resolution monitoring, and offering precise insights for both human and robotic actors. Recent developments in deep learning also offer competitive advantages for real-world applications, including generalization to unknown data [2–4] and model optimization for quick inference on low-power embedded hardware [5,6]. Simultaneously, advancements in service robotics have made it possible for self-governing mobile agents to assume the role of artificial intelligence perception systems and collaborate with them to complete intricate tasks in unstructured settings [7,8]. Among the most researched uses are row-based crops, which account for about 75% of all planted acres of agriculture in the United States [9]. The research involved in this scenario includes localization [10], path planning [11], navigation [12,13], monitoring [14], harvesting [15], spraying [16], and vegetative assessment [17,18]. It can be especially difficult when line-of-sight obstructions or adverse weather prevent traditional localization techniques like GPS from achieving the required precision. This is evident in dense tree canopies, as demonstrated in Fig. 1, which depicts a simulated pear orchard.

Previous works have proposed position-agnostic vision-based navigation algorithms for row-based crops, as discussed in Section 2. This work represents an extended version of a research presented in [19], tackling a more challenging scenario in which dense canopies partially or totally cover the sky, and the GPS signal is very weak. We design a navigation algorithm based on semantic segmentation that exploits

☆ This work has been developed with the contribution of the Politecnico di Torino Interdepartmental Centre for Service Robotics (PIC4SeR).
☆☆ This paper is part of the project NODES which has received funding from the MUR – M4C2 1.5 of PNRR funded by the European Union - NextGenerationEU (Grant agreement no. ECS00000036).

* Corresponding author at: Department of Electronics and Telecommunications, Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino, 10129, TO, Italy.
*E-mail addresses:* alessandro.navone@polito.it (A. Navone), mauro.martini@polito.it (M. Martini), marco.ambrosio@polito.it (M. Ambrosio), andrea.ostuni@polito.it (A. Ostuni), simone.angarano@polito.it (S. Angarano), marcello.chiaberge@polito.it (M. Chiaberge).

https://doi.org/10.1016/j.robot.2024.104854

Available online 8 November 2024
0921-8890/© 2024 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

**Fig. 1.** The proposed SegMin and SegMinD algorithms allow to precisely guide an autonomous mobile robot through a dense tree row solely using an RGB-D camera. An arched hedge is shown on the left, and a pergola vineyard row on the right.

visual perception to estimate the center of the crop row and align the robot trajectory to it. The segmentation masks are predicted by a deep learning model designed for real-time efficiency and trained on realistic synthetic images. The proposed navigation algorithm improves on previous works being adaptive to different terrains and crops, including dense canopies of different shapes. The experimentation conducted on the real field has been extended to previously unseen, challenging plant row conditions. Whenever possible, we compare our solution with previous state-of-the-art methodologies, moving a first fundamental step forward towards a common testing benchmark for autonomous visual navigation in row fields, despite the limits imposed by different experimental conditions and unavailable open source implementation of other methods. We demonstrate that the proposed navigation systems SegMin and SegMinD are effective and adaptive to numerous scenarios, previously not considered in literature.

The main contributions of this work can be summarized as follows:

- Two variants of a novel approach for segmentation-based autonomous navigation in tall crops, designed to tackle challenging and previously uncovered scenarios;
- Testing of the resulting guidance algorithm on previously unseen plant rows scenarios such as high orchards trees, pergola vineyards, and an arched hedge of plants.
- Training of an efficient segmentation neural network with synthetic multi-crop data only, proving the fast inference and generalization properties of the Deep Learning pipeline;
- A comparison of the new methods with state-of-the-art solutions on simulated and real vineyards, demonstrating an enhanced general and robust performance.

Both source code[1] and dataset[2] used to train our segmentation model have been publicly released.

The next sections are organized as follows: Section 2 briefly describes previous approaches presented in the literature to tackle visual-based navigation in row-based crops. Section 3 presents the proposed deep-learning-based control system for vision-based position-agnostic autonomous navigation in row-based crops, from the segmentation model to the controller. Section 4 describes the experimental setting and reports the main results for validating the proposed solution divided by sub-system. Finally, Section 5 draws conclusive comments on the work and suggests interesting future directions.

[1] https://github.com/PIC4SeR/SegMinNavigation.git
[2] https://pic4ser.polito.it/AgriSeg/

## 2. Related works

The competitive advantage of GPS-free visual approaches has been the subject of recent investigation within the field of service robotics. GPS-based localization and navigation have constituted the most widely applied strategy for outdoor and, consequently, agricultural robots [20]. Indeed, in outdoor scenarios, a costly GNSS positioning system equipped with RTK correction can provide a highly accurate localization for autonomous navigation, thereby enhancing the acquisition of geo-referenced data [21]. However, the quality of GNSS signal can be negatively affected by environmental features typical of row-based crops such as the presence of thick canopies, which lead to multi-path reflections and signal obstruction, and bad weather, which may hinder the signal accuracy [22–24].

Given its high resolution and large field of view, position-tracking algorithms based on LiDaR point clouds can provide a valuable alternative to GNSS-based autonomous navigation. 2D planar point clouds have been used in simple tests with only trunks by [25], although 2D scans are not sufficient to detect plants and canopies with complex shapes. On the other hand, 3D LiDARs with multiple layers demonstrate better performance in orchards navigation [26–28]. Nevertheless, LiDaR sensors present significantly high costs, limiting the large-scale spreading of agricultural vehicles on the market [29,30].

Vision-based navigation has emerged as a viable solution in highly unstructured and variable scenarios such as those encountered in orchards and crops, where high precision is required. The potential of machine vision in this field has been the subject of extensive research, largely due to the low cost of RGB-D cameras and the richness of the visual information that can be obtained [31]. Visual navigation encompasses a wide range of sensors, including monocular sensors [32], which are often employed for color detection and edge detection [33], and stereo sensors, which consist of more than one monocular camera and are able to capture three-dimensional and depth information, offering richer features for crop navigation [34,35].

In recent years, map-free navigation algorithms have gained importance as a prominent area of research to overcome the effort of building an accurate map of each specific field without a reliable localization signal. In row-based crops, the geometrical structure of the plants can be directly leveraged for autonomous navigation and data collection. Computer vision algorithms can be adopted in different fashions to extract from the image the necessary information to keep the automated vehicle at the center of the row while traversing the field. For instance, a vision-based approach was proposed in [36] using mean-shift clustering and the Hough transform to segment RGB images

and generate the optimal central path. Other works have also employed the Hough transform, as evidenced by the approaches proposed in [37,38]. In [37], the Hough matrix and Random Sample Consensus (RANSAC) were used to extract the navigation path, while in [38], a prediction-point Hough transform was utilized, comprising the steps of intercepting the area of interest, image segmentation, navigation point extraction, and navigation path fitting. More recently, Otsu's thresholding technique has been adopted by [39] to segment the sky, subsequently identifying the center of the row relying on sensitivity to light variation.

Alternative vision sensors have also been explored. Among them, [40] use multispectral images with a method of thresholding and filtering on the green channel to obtain meaningful features for autonomous navigation; in [41], drastic changes in the statistical distribution of points captured by a depth camera are exploited for row end detection; in [42], 3D vision has been merged with LiDaR and ultrasonic data.

Recently, deep learning approaches have been successfully applied to the task. [43,44] proposed a classification-based approach in which a model predicts the discrete action to perform. The detection of key objects in the scene can also represent the guiding principle of a visual controller. In their work, [45] proposed an end-to-end vision-based autonomous navigation stack based on a multi-task network. This approach allows for the simultaneous detection of tree trunks, obstacles, and traversable areas, with the detected trunks then utilized to determine the center and end of the row. The detection algorithm is integrated with real-time path planning and motion control, resulting in the generation of a short-term occupancy map and a global planner that determines the turning point to switch between rows and generates linear and angular velocities. Furthermore, also [46,47] rely on trunk detection. The former employs a method that combines vanishing point estimation with the average position of the two closest base trunk detections to estimate the angular orientation. The latter utilizes the least-squares algorithm to extract a navigation line.

In contrast, [48,49] proposed a proportional controller to align the robot to the center of the row using heatmaps of the scene first and segmented images in the latest version. A detailed analysis of the most recent literature reveals that semantic segmentation has definitely emerged as a reliable computer vision task to extract relevant information to navigate in the field. Indeed, [50,51] relied on an improved version of the DeepLabv3 model to segment the path comprised between the two plant rows and extract useful information to extract the trajectory at the center of the path. Similarly, [52] proposed a neural network model based on Unet and SegNet to detect the path between the plants and estimate the central trajectory from its edge.

The construction of a reliable neural network model necessitates the availability of a substantial quantity of data, however, the process of labeling RGB images has a considerable cost in terms of time and human effort. To mitigate this necessary condition of Deep Learning models, [53] proposed a learning-based system capable of estimating the path traversability heatmap from RGB-D images without the need for any human annotation. However, to achieve this, the automatic annotation pipeline relies on accurate GNSS ground truth data to predict the future poses of the robot along the path. Differently, an unsupervised learning methodology was introduced in [13] comprising an end-to-end controller based on Deep Reinforcement Learning and depth images. A potential solution to the generation of accurate labeled data in a short time is the exploitation of synthetic models and datasets, captured from simulated scenarios. Recent studies show that the resulting Sim2Real gap derived from the usage of synthetic images can be widely mitigated by combining realistic textures with data diversification, augmentation, and advanced generalization algorithms [4,54,55].

Overall, analyzing the current picture of GPS-free and map-less vision-based autonomous navigation in row-based crops, it emerged how, in both classic computer vision and deep learning methods, semantic segmentation results be a robust solution to obtain information about the geometry of the surrounding context. In fact, several methods have shown how, once the segmentation mask is obtained, the extraction of the central path to be followed is rather straightforward. However, an explicit solution comparison is hindered by the fact that trained models and results are tailored to the specific testing conditions on the field, which may present different challenges and features due to different plant types, terrain, and sky conditions. Some methods based on classic computer vision may show a stronger generalization capability when it comes to distinguishing high-contrast regions and strong and predictable features. However, they may fail in conditions where background conditions are not as expected. For example, in the case of [39], the presence of elements in the background may cause a different estimation of the sky region, leading to an erroneous evaluation of the trajectory. Deep learning methods can offer a wider generalization capability but strongly rely on training data, which may be hard to collect and label manually. Another limiting factor for several methods proposed is the visibility of the sky, which in some canopies is covered, leading to the systematic failure of some methods such as [39] or [44], which rely on sky visibility. The case that the trunk is not visible due to vegetation coverage may lead to the failure of methods such as [45–47]. In this work, we aim to propose an improved method to enhance such limitations, generalizing to the plant's presence only these strict environmental requirements. Nevertheless, creating a common benchmark and comparing existing methods is hindered by the non-availability of the open-source code of several works. For this reason, we release both the source code and the synthetic dataset used to train our models.

## 3. Methodology

To navigate high-vegetation orchards and arboriculture fields, this work provides a real-time control algorithm with two variations, which enhances the method described in [49]. The proposed method completely avoids employing GPS localization, which can be less accurate due to signal reflection and mitigation due to high and thick vegetation. Therefore, our algorithms consist of a straightforward operating principle, which exclusively employs RGB-D data and processes it to obtain effective position-agnostic navigation. It can be summarized in the following four steps:

1. Semantic segmentation of the RGB frame, with the purpose of identifying the relevant plants in the camera's field of view.
2. Addition of the depth data to the segmented frame to enhance the spatial understanding of the surrounding vegetation of the robot.
3. Searching for the direction towards the end of the vegetation row, given the previous information.
4. Generation of the velocity commands for the robot to follow the row.

However, the two suggested approaches only vary in steps 2 and 3, where they utilize depth frame data and generate the robot's desired direction. Conversely, the segmentation technique 1 and the command generation 4 are executed in a similar manner. A visual depiction of the proposed pipeline is illustrated in Fig. 2.

The first step of the proposed algorithm, at each time instant $t$ consists in acquiring an RGB frame $\mathbf{X}^t_{rgb}$ and a depth frame $\mathbf{X}^t_d$, where $\mathbf{X}^t_{rgb} \in \mathbb{R}^{h\times w\times c}$ and $\mathbf{X}^t_d \in \mathbb{R}^{h\times w}$. In both cases, $h$ represents the frame height, $w$ represents the frame width, and $c$ is the number of channels. The RGB data received is subsequently inputted into a segmentation neural network model $H_{seg}$, yielding a binary segmentation mask that conveys the semantic information of the input frame.

$$\hat{\mathbf{X}}^t_{seg} = H\left(\mathbf{X}^t_{rgb}\right) \tag{1}$$

where $\hat{\mathbf{X}}^t_{seg}$ is the obtained segmentation mask.

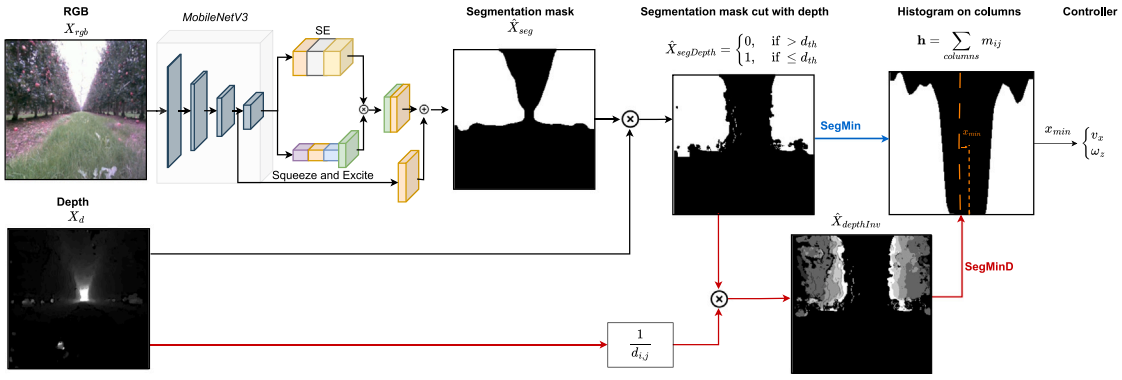



**Fig. 2.** Scheme of the overall proposed navigation pipeline. The RGB image is fed into the segmentation network, thus the predicted segmentation mask $\hat{\mathbf{X}}^t_{seg}$ is refined using the depth frame to obtain $\hat{\mathbf{X}}^t_{segDepth}$. The blue arrow refers to the SegMin variant, and red arrows refer to the SegMinD variant to compute the sum histogram over the mask columns. Images are taken from navigation in the tall trees simulation world. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Furthermore, the segmentation masks from the previous $N$ time instances, ranging from $t - N$ to $t$, are combined to enhance the robustness of the information.

$$\hat{\mathbf{X}}^t_{CumSeg} = \bigcup_{j=t-N}^{t} \hat{\mathbf{X}}^j_{seg} \tag{2}$$

where, $\hat{\mathbf{X}}^t_{CumSeg}$ denotes the cumulative segmentation mask, and the symbol $\bigcup$ signifies the logical bitwise $OR$ operation applied to the last $N$ binary frames.

Moreover, the depth map $\mathbf{X}^t_d$ is employed to assess the segmented regions between the camera position and a specified depth threshold $d_{th}$. This process helps eliminate irrelevant information originating from distant vegetation, which has no bearing on controlling the robot's movement.

$$\hat{\mathbf{X}}^t_{segDepth\ \substack{i=0,\dots,h \\ j=0,\dots,w}}(i,j) = \begin{cases} 0, & \text{if } \hat{\mathbf{X}}^t_{CumSeg(i,j)} \cdot \hat{\mathbf{X}}^t_{d(i,j)} > d_{th} \\ 1, & \text{if } \hat{\mathbf{X}}^t_{CumSeg(i,j)} \cdot \hat{\mathbf{X}}^t_{d(i,j)} \leq d_{th} \end{cases} \tag{3}$$

where, $\hat{\mathbf{X}}_{segDepth}$ represents the resultant intersection of the cumulative segmentation frame and the depth map, restricted to a distance threshold of $d_{th}$.

From this point forward, the proposed algorithm diverges into two variants, namely, *SegMin* and *SegMinD*, as elaborated in 3.1 and 3.2, respectively.

### 3.1. SegMin

The initial variant refines the methodology introduced in [49]. Following the segmentation mask processing, a column-wise summation is executed, generating a histogram $\mathbf{h} \in \mathbb{R}^w$ that characterizes the vegetation distribution along each column as in the following formula:

$$\mathbf{h}_j = \sum_{i=1}^{h} \hat{\mathbf{X}}_{segDepth(i,j)} \tag{4}$$

where $i = 0, \dots, w$ is the index along the vertical direction of each frame column.

Subsequently, a moving average is applied to this histogram using a window of size $n$ to enhance robustness by smoothing values and mitigating punctual noise from previous passes. In an ideal scenario, the minimum value $x_h$ in this histogram corresponds to regions with minimal vegetation, effectively pinpointing the central path within the crop row. If multiple global minima are identified, indicating areas with no detected vegetation, the mean of these points is calculated and considered as the global minimum. This approach ensures a more reliable identification of the continuation of the row, accommodating variations in the vegetation distribution.

### 3.2. SegMinD

The second proposed methodology presents a variation of the earlier algorithm tailored specifically for wide rows featuring tall and dense canopies. In such scenarios, the initial algorithm might encounter challenges in determining a clear global minimum, as the consistent presence of vegetation above the robot complicates the interpretation. This variant addresses this issue by incorporating a multiplication operation between the previously processed segmentation mask and the normalized inverted depth data.

$$\hat{\mathbf{X}}^t_{depthInv} = \hat{\mathbf{X}}^t_{segDepth} \odot \left(1 - \frac{\mathbf{X}^t_d}{d_{th}}\right) \tag{5}$$

where, $\hat{\mathbf{X}}^t_{depthInv}$ is the outcome of an element-wise multiplication, denoted by $\odot$, involving the binary mask $\hat{\mathbf{X}}^t_{segDepth}$ and the depth frame $\hat{\mathbf{X}}^t_d$ that has been normalized over the depth threshold $d_{th}$. Similar to the previous scenario, a column-wise summation is executed to derive the array $\mathbf{h}$, followed by a smoothing process using a moving average.

This introduced modification serves a crucial purpose by allowing elements closer to the robot to exert a more significant influence, thereby enhancing the algorithm's ability to discern the direction of the row.

The different sum histograms obtained with SegMin and SegMinD are directly compared in Fig. 3, showing the sharper trend and the global minimum isolation obtained, including the depth values.

### 3.3. Segmentation network

A prior study on crop segmentation in real-world conditions provided the neural network design that was chosen [49]. Fig. 4 illustrates its entire architecture and its primary benefit is its ability to leverage rich contextual information from the image at a lower computing cost.

A MobileNetV3 backbone makes up the network's initial stage, which is designed to efficiently extract the visual features from the input image [56]. With squeeze-and-excitation attention sub-modules [57], it is comprised of a series of inverted residual blocks [58]. They increase the amount of channel features while gradually decreasing the input image's spatial dimensions.

It is succeeded by a Lite R-ASPP (LR-ASPP) module [59], an enhanced and condensed variant of the Atrous Spatial Pyramid Pooling module (R-ASPP) that upscales the extracted features via two parallel branches. The first lower the spatial dimension by 1/16 by applying a Squeeze-and-Excite sub-module to the final layer of the backbone. To modify the number of channels $C$ to the output segmentation map, a channel attention weight matrix is produced, multiplied by the unpooled features, and then upsampled and fed through a convolutional layer. The second branch takes characteristics from an earlier stage of

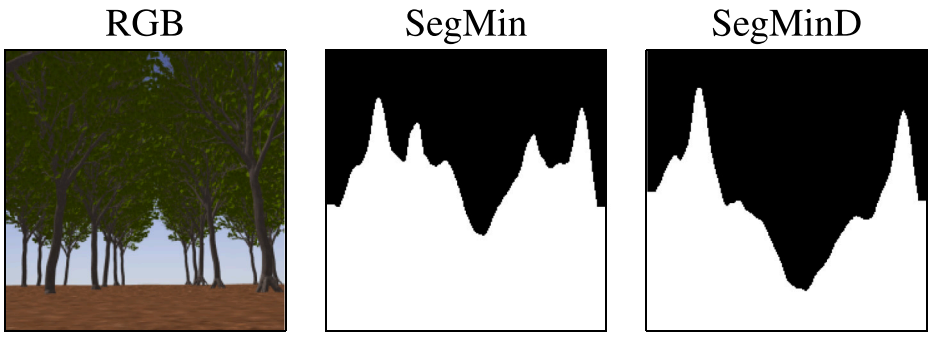



**Fig. 3.** Contrasting the histograms produced by the two distinct algorithms, considering the RGB frame on the right, reveals that SegMinD provides a more defined and less ambiguous global minimum point.

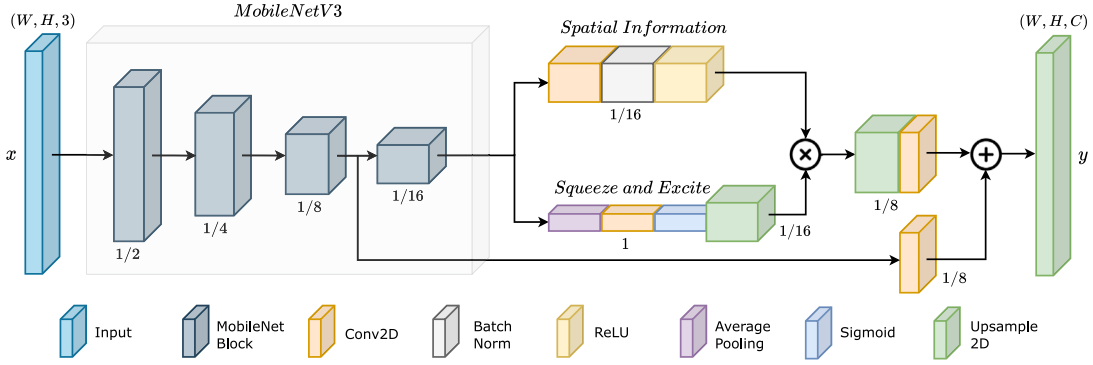



**Fig. 4.** The Deep Neural Network utilized in this study features a backbone of MobileNetV3 and an LR-ASPP head, as detailed in [56]. The spatial scaling factor of the features in comparison to the input size is provided beneath each block.

the backbone, which reduces the spatial dimension by 1/8, and adds them to the output of the upsampling step, mixing lower-level and higher-level patterns in the data.

The network's input has a dimensionality equal to $W \times H \times 3$, while the segmented output is equal to $W \times H$.

Furthermore, the neural network's output values are scaled between 0 and 1 using a sigmoid function, as this work primarily focuses on the semantic segmentation of plant rows.

The usual cross-entropy loss between the ground-truth label $y$ and the anticipated segmentation mask is used to train the DNN:

$$L_{\mathrm{CE}}(y, \hat{y}) = -\sum_{i=1}^{N} y_i \cdot log(\hat{y}_i) \tag{6}$$

which for binary segmentation becomes a simple binary cross-entropy loss.

During both the validation and testing phases, the DNN performance is evaluated through an intersection over unit (IoU) metrics:

$$mIoU(\theta) = \frac{1}{N}\sum_{i=0}^{N}\left(1 - \frac{\hat{X}^i_{seg} \cap X^i_{seg}}{\hat{X}^i_{seg} \cup X^i_{seg}}\right) \tag{7}$$

where $X^i_{seg}$ is the ground truth mask, $\hat{X}^i_{seg}$ is a predicted segmentation mask, and $\theta$ is the vector representing the network parameters. Since there are only plants as the target class of interest, $N$ in the IoU computation always equals 1. The model is trained on the AgriSeg synthetic dataset [4,55].[3] Further details on the training strategy and hyperparameters are provided in Section 4.

### 3.4. Robot heading control

The goal of the controller pipeline is to maintain the mobile platform at the center of the row, which, in this study, is equated to aligning the row center with the middle of the camera frame. Consequently, following the definition in the preceding step, the minimum of the histogram should be positioned at the center of the frame width. The distance $d$ from the frame center to the minimum is defined as:

$$d = x_h - \frac{w}{2} \tag{8}$$

The generation of linear and angular velocities is accomplished using custom functions, mirroring the approach employed in [60].

$$v_x = v_{x,max}\left[1 - \frac{d^2}{\left(\frac{w}{2}\right)^2}\right] \tag{9}$$

$$\omega_z = -k_{\omega_z} \cdot \omega_{z,max} \cdot \frac{d^2}{w^2} \tag{10}$$

where, $v_{x,max}$ and $\omega_{z,max}$ represent the maximum attainable linear and angular velocities, and $k_{\omega_z}$ serves as the angular gain controlling the response speed. To mitigate abrupt changes in the robot's motion, the ultimate velocity commands $\bar{v}x$ and $\bar{\omega}z$ undergo smoothing using an Exponential Moving Average (EMA), expressed as:

$$\begin{bmatrix}\bar{v}^t_x \\ \bar{\omega}^t_z\end{bmatrix} = (1-\lambda)\begin{bmatrix}\bar{v}^{t-1}_x \\ \bar{\omega}^{t-1}_z\end{bmatrix} + \lambda\begin{bmatrix}v^t_x \\ \omega^t_z\end{bmatrix} \tag{11}$$

where, $t$ represents the time step, and $\lambda$ stands for a selected weight.

## 4. Experiments and results

### 4.1. Segmentation network training and evaluation

We train the crop segmentation model using a subset of the AgriSeg synthetic segmentation dataset [4,54]. In particular, for the pear trees and apple trees, we train on generic tree datasets in addition to pear and apples; for vineyards, we train on vineyard and pergola vineyards (note that the testing environments are different from the ones from

[3] https://pic4ser.polito.it/AgriSeg

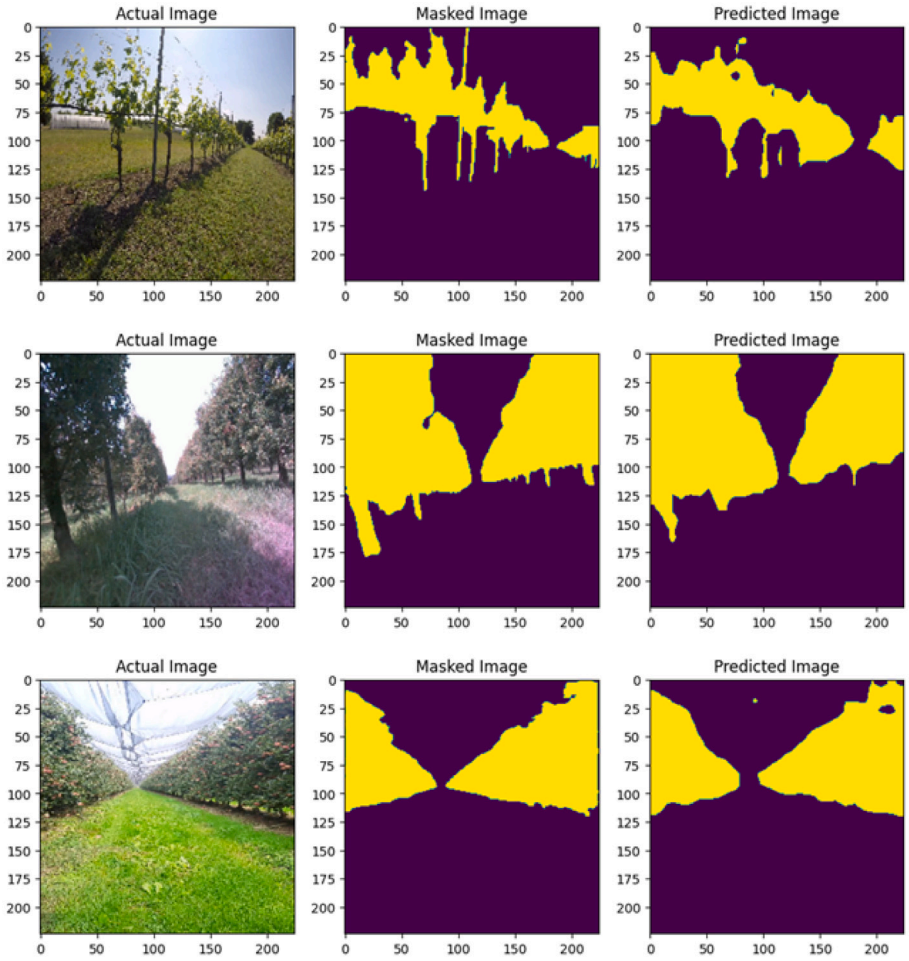



**Fig. 5.** Test of semantic segmentation DNN on real-world test samples from vineyard (top), pear trees (middle) and apple trees (bottom) fields. For each crop, RGB input image (left), ground truth mask (center) and the predicted mask (right) are reported.

**Table 1**
Semantic Segmentation results on real images in different crop fields. For each crop, a model has been trained on synthetic data, only using 100 additional real images containing miscellaneous crops different from the test set.

| Model | Real test mIoU | Train data | Real test data |
|---|---|---|---|
| Vineyard | **0.6950** | 13 840 | 500 |
| Apples | **0.8398** | 15 280 | 210 |
| Pear | **0.8778** | 7980 | 140 |

which the training samples are generated). Only 100 miscellaneous real images of different crop types are added to the training dataset in all the cases. Thanks to the high-quality rendering of the AgriSeg dataset, this small amount of real images is sufficient to reach general and robust performance in real-world conditions. In both cases, the model is trained for 30 epochs with Adam optimizer and learning rate $3 \times 10^{-4}$. We apply data augmentation by randomly applying cropping, flipping, grayscaling, and random jitter to the images. Our experimentation code is developed in Python 3 using TensorFlow as the deep learning framework. We train models starting from ImageNet pretrained weights, so the input size is fixed to $(224 \times 224)$. All the training runs are performed on a single Nvidia RTX 3090 graphic card.

Table 1 reports the results obtained testing the trained segmentation DNN on real images in terms of mean Intersection over Union (IoU), as defined in Eq. (7). Fig. 5 also shows some qualitative results on sample images collected on the field during the test campaign.

### 4.2. Simulation environment

The proposed control algorithm underwent testing using the Gazebo simulation software.[4] Gazebo was chosen due to its compatibility with ROS 2 and its ability to integrate plugins simulating sensors, including cameras. A Clearpath Jackal model was employed to evaluate the algorithm's performance. The URDF file from Clearpath Robotics, containing comprehensive information about the robot's mechanical structure and joints, was utilized. In the simulation, an Intel Realsense D435i plugin was employed, placed 20 cm in front of the robot's center, and tilted upward by 15°: this configuration enhanced the camera's visibility of the upper branches of trees.

The assessment of the navigation algorithm took place in four customized simulation environments, each designed to mirror distinct agricultural scenarios. These environments included a conventional vineyard, a pergola vineyard characterized by elevated vine poles and shoots above the rows, a pear field populated with small-sized trees, and a high-tree field where the canopies interweave above the rows. Each simulated field features varied terrains, replicating the irregularities found in real-world landscapes. Comprehensive measurements for each simulation world can be found in Table 2.

In the experimental phase of this study, we adopted frame dimensions of $(h, w) = (224, 224)$, matching the input and output sizes of the neural network model. Moreover, the number of previous segmentation

[4] https://gazebosim.org

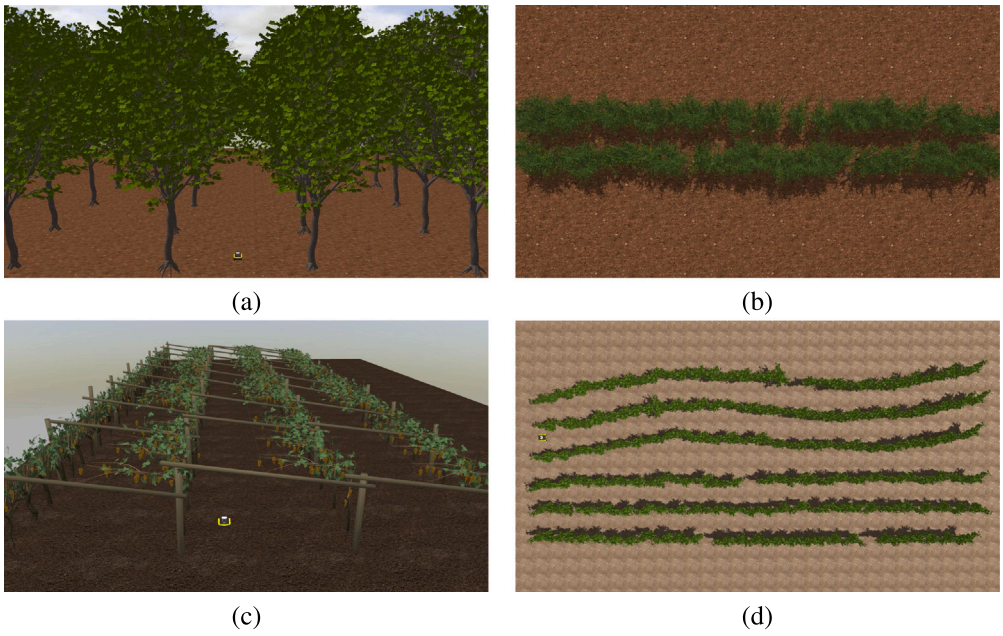



**Fig. 6.** Gazebo simulated environments were employed to assess the SegMin approach in various crop rows of significance, including wide rows with high trees (a), a slender row of pear trees (b), an asymmetric pergola vineyard with irregular rows (c), and both straight and curved vineyard rows (d). For the latter scenario, the evaluations were conducted in both the second row from the top and the second row from the bottom.

**Table 2**
Dimensions of various simulated crops indicate the average values for the distance between rows, the spacing between plants within a row, and the heights of the plants.

| Gazebo worlds | Rows distance [m] | Plant distance [m] | Height [m] |
|---|---|---|---|
| Common vineyard | 1.8 | 1.3 | 2.0 |
| Pergola vineyard | 6.0 | 1.5 | 2.9 |
| Pear field | 2.0 | 1.0 | 2.9 |
| High trees field | 7.0 | 5.0 | 12.5 |

masks $N$ to be combined to enhance robustness, as in Eq. (2), is set to 3. The maximum linear velocity was set to $v_{x,max} = 0.5$ m/s, and the maximum angular velocity was capped at $\omega_{z,max} = 1$ rad/s. The angular velocity gain, denoted as $\omega_{z,gain}$, was fixed at 0.01, and the Exponential Moving Average (EMA) buffer size was set to 3. Additionally, the depth threshold was adjusted based on the specific characteristics of different crops.

Specifically, it has been empirically set at 5 m for vineyards, raised to 8 m for pear trees and pergola vineyards, and further increased to 10 m for tall trees, taking into account the average distance from the rows in various fields.

### 4.3. Navigation results in simulation

The comprehensive evaluation of the SegMin navigation pipeline and its variant, SegMinD, took place in realistic crop fields within a simulation environment, employing pertinent metrics for visual-based control without the need for precise robot localization, aligning with methodologies from prior studies [13,49]. The camera frames were published at a frequency of 30 Hz, with inference conducted at 20 Hz and velocity commands from controllers published at 5 Hz. The evaluation utilized the testing package from the open-source PIC4rl-gym[5] in Gazebo [61]. The chosen metrics aimed to assess the navigation effectiveness, measured by clearance time and precision, involving a quantitative comparison of obtained trajectories with a ground truth trajectory using Mean Absolute Error (MAE) and Mean Squared Error (MSE). Ground truth trajectories were computed by averaging interpolated poses of plants within rows. In the case of an asymmetric pergola vineyard, a row referred to the portion without vegetation on top, as depicted in Fig. 6(c). The algorithms' response to terrain irregularities and row geometries was also studied, encompassing significant kinematic information about the robot. The evaluation considered the cumulative heading average $\gamma$ [rad] along the path, mean linear velocity $v_{avg}$ [m/s], and standard deviation of angular velocity $\omega_{stddev}$ [rad/s]. These metrics provided insights into how well the algorithms maintained the robot's correct orientation, with the mean value of $\omega$ consistently approaching zero due to successive orientation corrections.

The complete set of results is outlined in Table 3. Each metric is accompanied by both the average value and standard deviation, reflecting the repetition of experiments in three runs on a 20 m long track within each crop row. The proposed method effectively addresses the challenge of guiding the robot through rows of trees with dense canopies, such as high trees and pears, even in the absence of a localization system. It also demonstrates proficiency in unique scenarios, like navigating through pergola vineyards. The presence of plant branches and wooden supports poses a challenge for existing segmentation-based solutions. These solutions, built on the assumption of identifying a clear passage by focusing solely on zeros in the binary segmentation mask [49], encounter limitations in our tested scenarios. In our result comparisons, we term this prior method as SegZeros, utilizing the same segmentation neural network for assessment.

The SegMin methodology, based on histogram minimum search, proves to be a resilient solution for guiding the robot through tree rows. The incorporation of depth inverse values as a weighting function in SegMinD enhances the algorithm's precision, particularly in navigating through challenging scenarios like wide rows (high trees) and curved rows (curved vineyard). Furthermore, these innovative methods exhibit competitive performance even in standard crop rows, where a clear passage to the end of the row is discernible in the mask without canopy interference. Compared to the previous segmentation-based baseline method, the histogram minimum approach significantly reduces navigation time and enhances trajectory precision in both straight and curved vineyard rows. On the other hand, the search for plant-free zero clusters in the map proves to be less robust and efficient, leading to undesired stops and an overall slower and more oscillating behavior

[5] https://github.com/PIC4SeR/PIC4rl_gym

**Table 3**
Navigation outcomes across diverse test fields were assessed using the SegMin, SegMinD, and the SegZeros segmentation-based algorithms. The evaluation employed metrics to gauge the efficacy of navigation, including clearance time, and assessed precision through Mean Absolute Error (MAE) and Mean Squared Error (MSE) by comparing the obtained path with the ground truth. Additionally, kinematic information about the robot's navigation was captured through the cumulative heading average $\gamma$[rad], mean linear velocity $v_{avg}$[m/s], and the standard deviation of angular velocity $\omega_{stddev}$[rad/s]. Notably, SegZeros proved impractical for scenarios involving tall trees, pear trees, and pergola vineyards.

| Test field | Method | Clearance [s] | MAE [m] | MSE [m] | Cum. $\gamma_{avg}$ [rad] | $v_{avg}$ [m/s] | $\omega_{stddev}$ [rad/s] |
|---|---|---|---|---|---|---|---|
| High Trees | SegMin | **40.41 ± 0.12** | 0.27 ± 0.01 | 0.08 ± 0.00 | 0.08 ± 0.00 | 0.49 ± 0.00 | 0.05 ± 0.00 |
| | SegMinD | 40.44 ± 0.51 | **0.17 ± 0.01** | **0.04 ± 0.00** | 0.05 ± 0.00 | 0.48 ± 0.01 | 0.06 ± 0.02 |
| Pear Trees | SegMin | **42.06 ± 1.23** | 0.03 ± 0.01 | **0.00 ± 0.00** | 0.01 ± 0.00 | 0.48 ± 0.00 | 0.11 ± 0.05 |
| | SegMinD | 42.26 ± 1.91 | **0.03 ± 0.02** | **0.00 ± 0.00** | 0.02 ± 0.00 | 0.48 ± 0.01 | 0.03 ± 0.00 |
| Pergola Vine. | SegMin | **40.86 ± 0.39** | **0.08 ± 0.01** | **0.01 ± 0.00** | 0.03 ± 0.02 | 0.48 ± 0.00 | 0.17 ± 0.02 |
| | SegMinD | 41.14 ± 0.33 | 0.10 ± 0.05 | 0.01 ± 0.01 | 0.03 ± 0.01 | 0.48 ± 0.00 | 0.20 ± 0.03 |
| Straight Vine. | SegMin | **50.51 ± 0.31** | **0.11 ± 0.00** | **0.01 ± 0.00** | 0.03 ± 0.00 | 0.49 ± 0.00 | 0.08 ± 0.01 |
| | SegMinD | 50.63 ± 0.28 | 0.11 ± 0.01 | 0.02 ± 0.00 | 0.03 ± 0.01 | 0.49 ± 0.00 | 0.09 ± 0.01 |
| | SegZeros | 53.69 ± 1.03 | 0.14 ± 0.03 | 0.02 ± 0.01 | 0.03 ± 0.0 | 0.46 ± 0.01 | 0.09 ± 0.01 |
| Curved Vine. | SegMin | 53.32 ± 0.25 | 0.12 ± 0.01 | 0.02 ± 0.00 | 0.04 ± 0.01 | 0.49 ± 0.00 | 0.09 ± 0.02 |
| | SegMinD | **51.44 ± 1.03** | **0.09 ± 0.01** | **0.01 ± 0.00** | 0.01 ± 0.00 | 0.48 ± 0.01 | 0.06 ± 0.01 |
| | SegZeros | 71.05 ± 27.13 | 0.11 ± 0.04 | 0.02 ± 0.01 | 0.05 ± 0.01 | 0.40 ± 0.13 | 0.11 ± 0.04 |

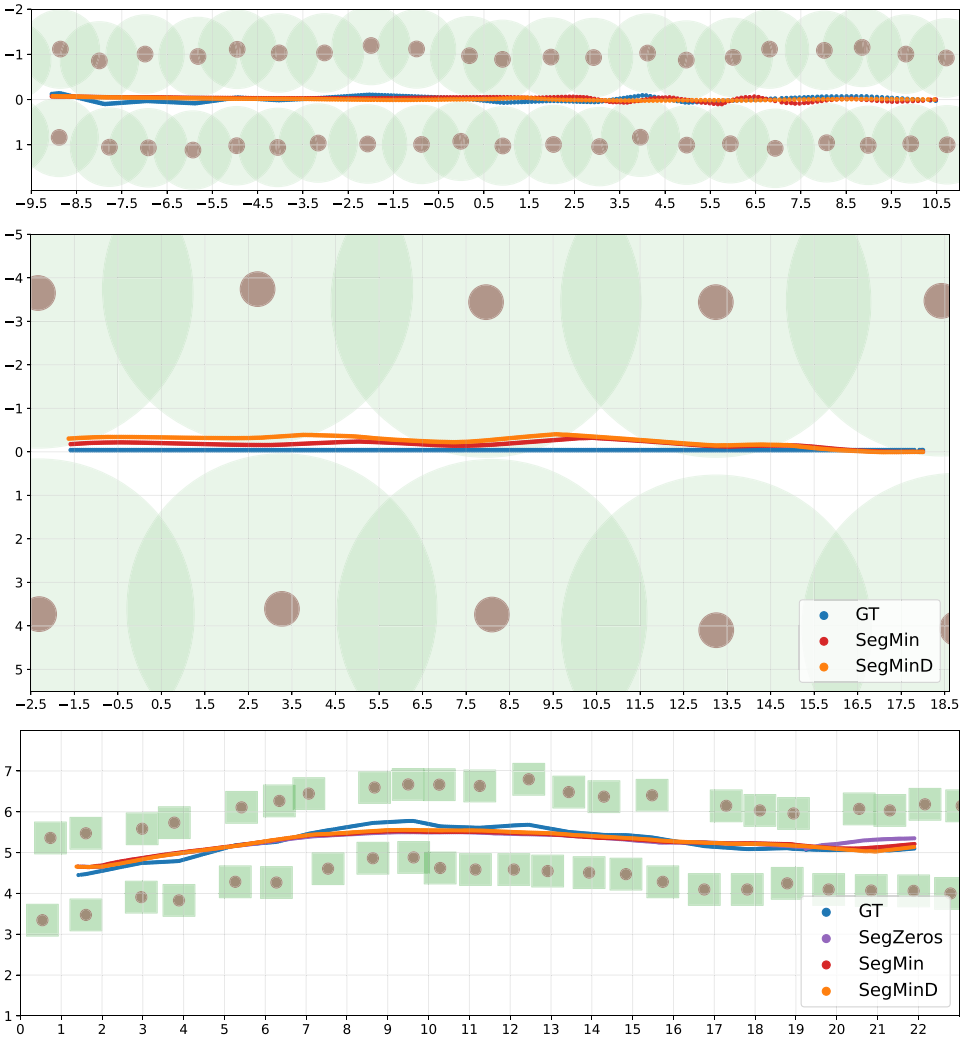



**Fig. 7.** Trajectories comparison between our proposed algorithms (SegMin and SegMinD) and the ground truth central path (GT): Pears (top), High Trees (center), Curved Vineyard (bottom). In the last graph, the trajectory generated with the SegZeros algorithm is also reported for comparison.

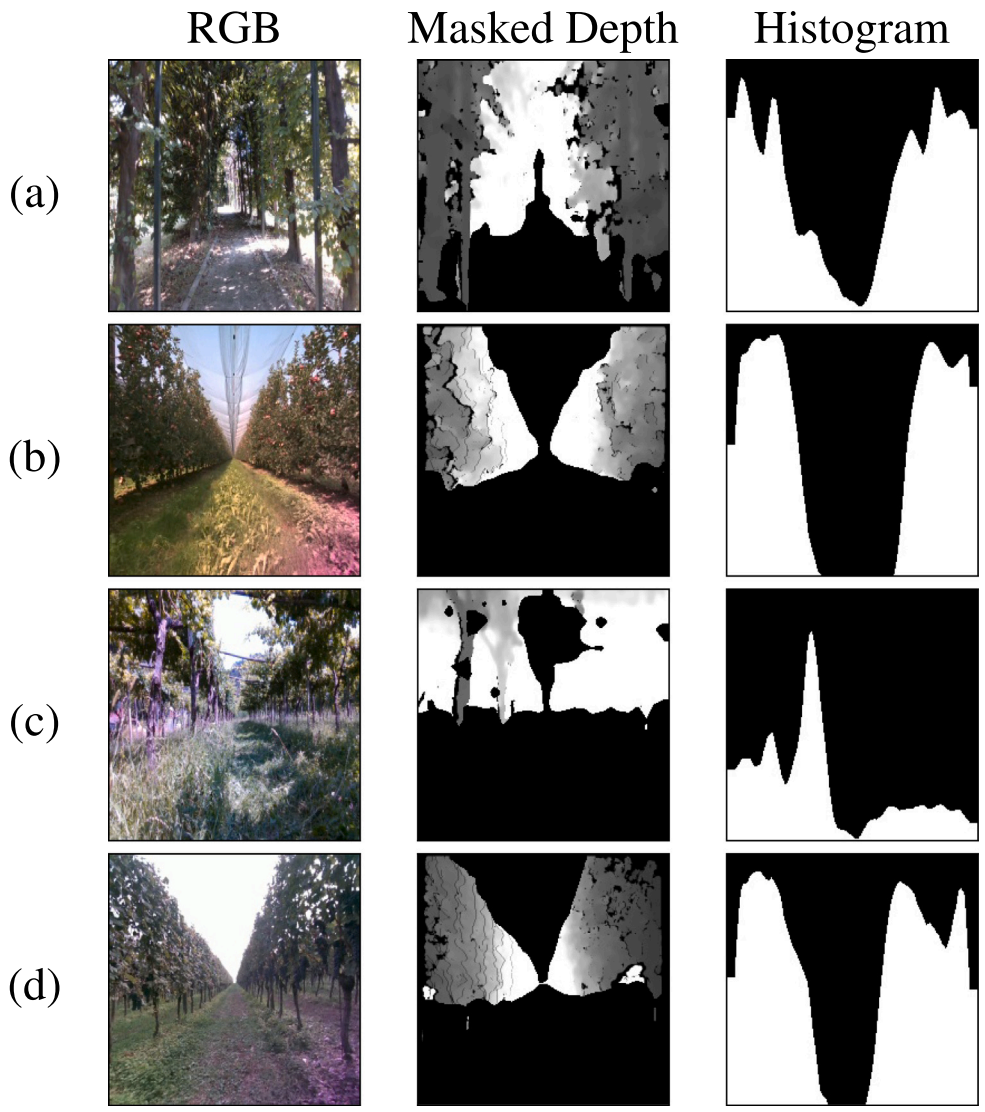



**Fig. 8.** Sample outputs of the proposed SegMinD algorithm for High Trees (arched) (a), Pear Trees (b), Pergola Vineyard (c), and Vineyard (d). Predicted segmentation masks are refined cutting values exceeding a depth threshold. The sum over mask columns provides the histograms used to identify the center of the row as its global minimum.

during navigation. Additionally, the standard deviation of angular velocity aligns with the results obtained, being smaller in cases where the trajectory is more accurate, while the cumulative heading exhibits larger values when the algorithms demonstrate increased reactivity.

However, the trajectories generated by the SegMin, SegMinD, and SegZeros algorithms are visually depicted in Fig. 7 within representative scenarios. These scenarios include a cluttered, narrow row featuring small pear trees, a wide row with high trees, and curved vineyards where the state-of-the-art SegZeros method is applied.

### 4.4. Navigation test on the field

The overall navigation pipeline of SegMin and its variant SegMinD are tested in real crop fields, evaluating the results with relevant metrics for visual-based control without precise localization of the robot, as done in previous works [13,49]. The robotic platform employed to perform the tests is a Clearpath Husky UGV equipped with a LiDAR Velodyne Puck VLP-16, an RGBD camera Realsense D455, an AHRS Microstrain 3DM-GX5 and a Mini-ITX computer with an Intel Core i7 processor and 16 GB of memory. The camera frames were captured at a rate of 30 Hz, inference was performed at 20 Hz, and velocity commands were published at 5 Hz. In this section's experiments, ground truth trajectory was unavailable due to the complexity and demanding nature of measurement, which requires sophisticated instruments for sufficient accuracy. Instead, the lateral displacement of the rover within the row was determined using point clouds from the LiDAR. Points were clustered to separate the two rows, then fitted by a straight line, followed by computing the shortest distance from the plants to the origin, i.e., the center of the robot, where the sensor is mounted, for both lanes. The AHRS measured the rover's heading and compared it with the average heading of the row obtained from satellite images, considering the tested rows are straight.

The performances of the proposed control have been evaluated on two different plant orchards, i.e., apples and pears, a straight vineyard, a pergola vineyard, and an arched hedge, where the canopies are very tight. The intermediate output of the algorithm can be seen in Fig. 8, where the RGB frames are alongside the respective segmentation mask merged with the depth data and the corresponding histogram. The performance metrics are reported in Table 4. The trajectories of the best test for each crop field and the comparison between the proposed algorithms, SegMin, SegMinD, and SegZeros, are represented in Figs. 9 and 10. Overall, the novel control laws can effectively solve the problem of guiding the robot through tree rows with thick canopies (high trees and pears) without a localization system in a real-world scenario.

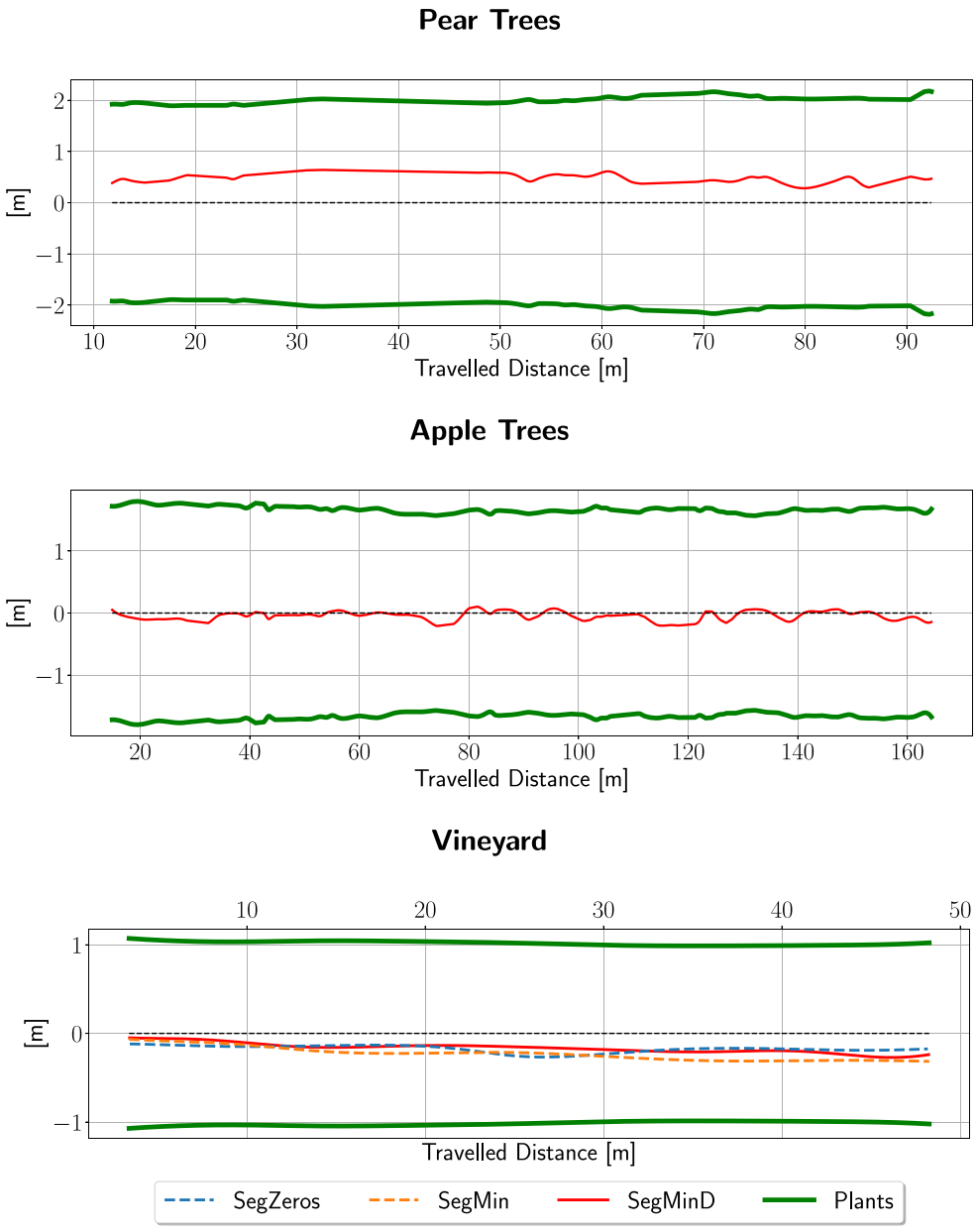



**Fig. 9.** Trajectory results of relevant tests performed on the field. In order from top to bottom: navigation in pear tree rows using SegMinD, navigation in apple tree rows using SegMinD, and trajectory comparison of all three algorithms in a vineyard row. Sudden drifts in orchards traversal are caused by fruits, small obstacles, and irregularity in the terrain.

**Table 4**
Navigation results of the real-world testing of the algorithms. SegMin and SegMinD have been tested in apple and pear orchards, a pergola vineyard and an arched hedge. A comparison has been performed between the algorithms SegZeros, SegMin and SegMinD in a vineyard row choosing the same parameters: depth threshold set to 8.0 $m$ and pixel-wise confidence to 0.7. The error regarding the pergola vineyard are evaluated considering the center of the lateral row where the sky is visible.

| | Algorithm | MAE [m] | RMSE [m] | Cum. $\gamma_{avg}$ [rad] | $\omega$ STD [rad/s] |
|---|---|---|---|---|---|
| Apple Trees | SegMin | 0.167 | 0.188 | −0.032 | 0.041 |
| | SegMinD | **0.072** | **0.091** | 0.107 | 0.068 |
| Pear Trees | SegMin | 0.465 | 0.473 | 0.030 | 0.129 |
| | SegMinD | **0.284** | **0.297** | 0.030 | 0.130 |
| Straight Vine. | SegZeros | 0.166 | **0.170** | −0.012 | 0.026 |
| | SegMin | 0.230 | 0.241 | −0.033 | 0.114 |
| | SegMinD | **0.160** | **0.170** | −0.114 | 0.101 |
| Pergola Vine. | SegMin | **0.012** | **0.002** | 0.085 | 0.062 |
| | SegMinD | 0.059 | 0.047 | 0.088 | 0.105 |
| Arched Hedge | SegMin | 0.054 | 0.087 | −0.018 | 0.043 |
| | SegMinD | **0.049** | **0.063** | −0.017 | 0.047 |

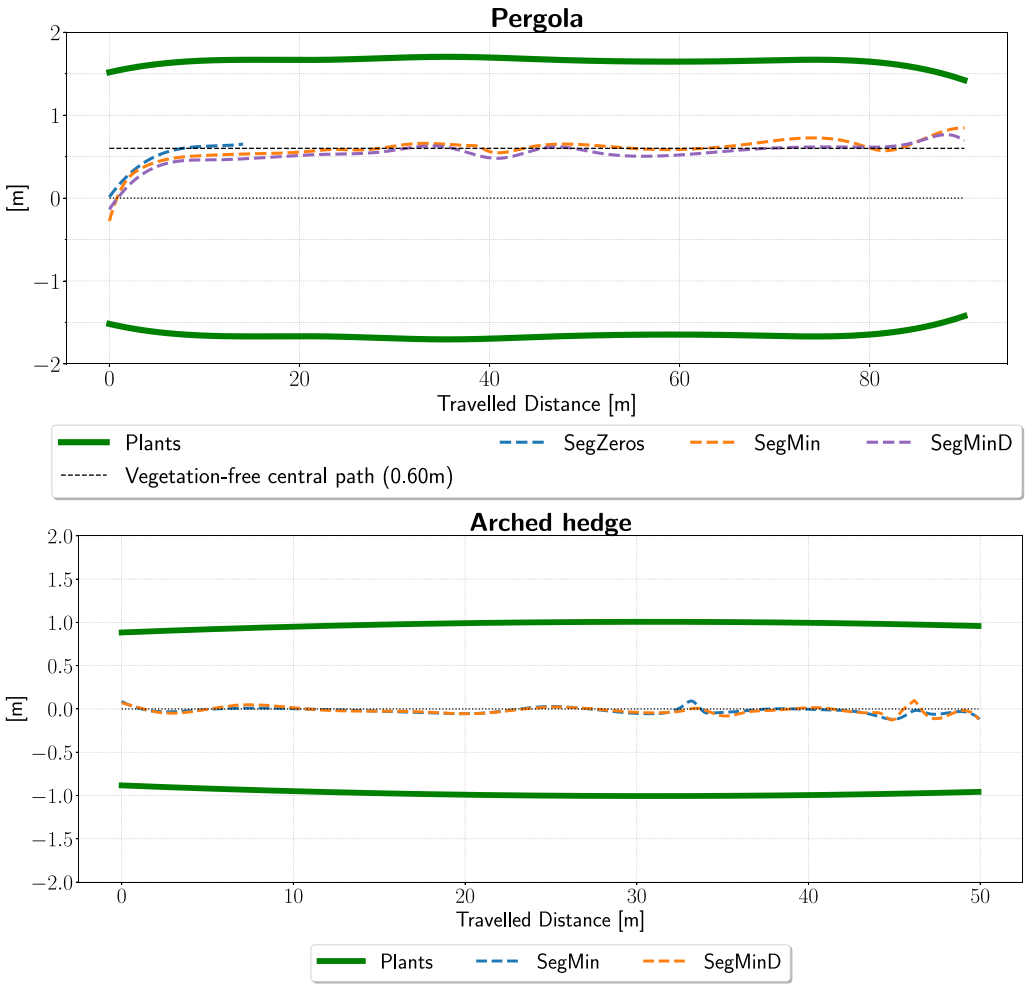



**Fig. 10.** Trajectory results of relevant tests performed on the field in case of crops where the sky is occluded. On the top, the case of pergola vineyards, where the robot is expected to navigate on the vegetation-free path, whose center is 0.6 m on the side of the lane center. The trajectory of SegZeros fails due to the vegetation coverage. On the bottom, the trajectory in the arched hedge scenario. Sudden drifts in orchards traversal are caused by small obstacles and irregularity in the terrain.

The algorithms SegMin and SegMinD demonstrate the ability to generalize to the common case without obstruction by canopies. As shown by the comparison performed in the vineyard, they obtained results in line with the existing SegZeros. The algorithms SegMin and SegMinD show their effectiveness in maintaining the robot on the desired central line, even recovering from strong disturbances. As can be noticed by the trajectories in pear and apple trees (top and central plots in Fig. 9), sudden drifts of the robot are caused by fruits, branches, stones, and disparate irregularity of the terrain. Those small obstacles cannot be precisely sensed and tackled with classic obstacle avoidance algorithms; hence, the resilience of the control algorithms to these external factors is crucial to keep the robot on track. Differently, in vineyard rows, grass and cleaner terrain induce smoother overall trajectories.

Moreover, the proposed SegMin and SegMinD algorithms demonstrate accurate performance even in the context of pergola vineyards and arched hedges, where vegetation partially or completely obscures the sky. It is noteworthy that in pergola vineyards, where the sky is visible from the side of the row, the robot follows a path where the sky is most visible, thereby maintaining its trajectory at the center of the vegetation gap. As illustrated in Fig. 10, the center of the vegetation-free portion of the row is estimated to be 0.6 m to the left of the row center. With more vegetation, this scenario would become analogous to the arched hedge, where the sky is entirely covered. In such cases, depth data combined with the segmentation mask plays a crucial role in estimating the continuation of the row and consequently, the evaluation of the desired trajectory. The deviation from the central trajectory that can be noticed in the bottom graph of Fig. 10, are due to gaps and irregularities in the hedge's vegetation.

The proposed algorithms have been compared also with a state of art scene segmentation solution proposed in [39], which is based on the Otsu thresholding algorithm. It was tested in the same conditions as the one described in the original paper, however some critical issues raised. In fact, the algorithm proposed in the paper relies on the shape of the sky visible by the robot's front camera, which allows the identification of the continuation of the row. However, elements in the background as hills, which were present in our testing scenario, did not allow a clear threshold identification. Moreover, the presence of bright plant supports misled the computer vision algorithm, which, identifying them as bright, estimated an untrue continuation of the row, causing the trajectory to divert from the desired one.

The comparison with methods based on deep learning requires the source code and the data used by the authors, or at least the trained model. In fact, deep neural networks trained with different hyperparameters and data will lead to a heavily diverse outputs, invalidating the experiments. The comparison with other recent methods based on the detection of the trunks, such as [45], or the detection of the

**Table 5**
Ablation study on SegMinD algorithm performance: relevant parameters of the segmentation control system are explored for a better understanding of their impact on the overall result. Three values of depth threshold and prediction confidence are selected for the ablation.

| Depth threshold | MAE [m] | RMSE [m] | Cum. $\gamma_{avg}$ [rad] | $\omega$ STD [rad/s] |
|---|---|---|---|---|
| Confidence 0.3 | | | | |
| 5.0 | 0.352 | 0.360 | −0.279 | 0.990 |
| 8.0 | 0.228 | 0.238 | −0.012 | 0.250 |
| 11.0 | 0.352 | 0.362 | 0.067 | 0.384 |
| Confidence 0.5 | | | | |
| 5.0 | 0.224 | 0.239 | 0.027 | 0.420 |
| 8.0 | 0.157 | 0.171 | −0.199 | 0.409 |
| 11.0 | 0.455 | 0.457 | 0.003 | 0.103 |
| Confidence 0.7 | | | | |
| 5.0 | 0.386 | 0.394 | 0.029 | 0.465 |
| **8.0** | **0.119** | **0.150** | **−0.040** | **0.125** |
| 11.0 | 0.396 | 0.402 | −0.016 | 0.359 |

path, such as [51,53] would have been beneficial for this work and for the research community. However, the original codes and datasets of the aforementioned papers are not publicly available, hence we invite researchers to share their solution and we leave the creation of a common benchmark for orchards navigation as future work.

Finally, an ablation study is carried out on a vineyard row to assess the impact of key parameters on the proposed control strategy within the novel SegMinD algorithm. Specifically, the study explores the effects of the depth image max distance and pixel-wise confidence threshold of the predicted segmentation mask. The findings, detailed in Table 5, indicate that a confidence level greater than 0.5 is required for achieving robust behavior, filtering mask portions with uncertain prediction. Indeed, results with a confidence level of 0.3 exhibit a high standard deviation in the angular velocity command. Regarding the depth image maximum distance, three values are tested: 5, 8, and 11 m. A low value of 5 m produces sub-optimal results compared to an intermediate value of 8 m. In this scenario, the noise in the segmentation has a more pronounced effect as the long-view geometry of the row is not considered in the computation of the histogram, including only close plants. Conversely, a high value for the depth threshold leads to inferior results due to the insufficient precision of the depth camera, resulting in artifacts that can compromise overall performance. In conclusion, the optimal outcome is achieved with a high confidence in the prediction and an intermediate depth threshold of 8 m.

## 5. Conclusion

In this work, we presented a novel method to guide a service-autonomous platform through crop rows where a precise localization signal is often occluded by vegetation. Trees rows represented an open problem in row crop navigation since previous works based on image segmentation or processing failed due to the presence of branches and canopies covering the free passage for the rover in the image. The proposed pipeline SegMin and SegMinD overcome this limitation by introducing a global minimum search on the sum histogram over the mask columns. The experiments conducted demonstrate the ability to solve the navigation task in wide and narrow tree rows and, nonetheless, the improvement in efficiency and robustness provided by our method over previous works in generic vineyards scenarios. Moreover, real-world tests proved the reliable generalization properties of the efficient semantic segmentation neural network trained with synthetic data only.

Future work will see the extension of the robot's capabilities to support agricultural tasks where more complex multi-objective behaviors are required, such as plant approach, box transport, and harvesting.

## CRediT authorship contribution statement

**Alessandro Navone:** Writing – original draft, Software, Methodology, Formal analysis, Conceptualization. **Mauro Martini:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Formal analysis, Conceptualization. **Marco Ambrosio:** Writing – original draft, Visualization, Validation, Software, Methodology, Data curation. **Andrea Ostuni:** Writing – original draft, Methodology, Data curation. **Simone Angarano:** Writing – original draft, Methodology, Investigation, Conceptualization. **Marcello Chiaberge:** Writing – review & editing, Funding acquisition.

## Declaration of competing interest

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

## Data availability

We have shared the code and the data link in the paper. The data is protected by a password for security reasons but will be given to anyone asking for it.

## References

[1] Z. Zhai, J.F. Martínez, V. Beltran, N.L. Martínez, Decision support systems for agriculture 4.0: Survey and challenges, Comput. Electron. Agric. 170 (2020) 105256.
[2] M. Martini, V. Mazzia, A. Khaliq, M. Chiaberge, Domain-adversarial training of self-attention-based networks for land cover classification using multi-temporal sentinel-2 satellite imagery, Remote Sens. 13 (13) (2021) 2564.
[3] S. Angarano, M. Martini, F. Salvetti, V. Mazzia, M. Chiaberge, Back-to-bones: rediscovering the role of backbones in domain generalization, Pattern Recognition 156 (2024) 110762.
[4] S. Angarano, M. Martini, A. Navone, M. Chiaberge, Domain generalization for crop segmentation with standardized ensemble knowledge distillation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 5450–5459.
[5] V. Mazzia, A. Khaliq, F. Salvetti, M. Chiaberge, Real-time apple detection system using embedded systems with hardware accelerators: An edge AI application, IEEE Access 8 (2020) 9102–9114.
[6] S. Angarano, F. Salvetti, V. Mazzia, G. Fantin, D. Gandini, M. Chiaberge, Ultra-low-power range error mitigation for ultra-wideband precise localization, in: Intelligent Computing: Proceedings of the 2022 Computing Conference, vol. 2, Springer, 2022, pp. 814–824.
[7] J. Holland, L. Kingston, C. McCarthy, E. Armstrong, P. O'Dwyer, F. Merz, M. McConnell, Service robots in the healthcare sector, Robotics 10 (1) (2021) 47.
[8] A. Eirale, M. Martini, L. Tagliavini, D. Gandini, M. Chiaberge, G. Quaglia, Marvin: An innovative omni-directional robotic assistant for domestic environments, Sensors 22 (14) (2022) 5261.
[9] D. Bigelow, A. Borchers, Major uses of land in the United States, 2012, Econ. Inform. Bull. Number 178 178 (1476-2017-4340) (2017) 69.
[10] W. Winterhalter, F. Fleckenstein, C. Dornhege, W. Burgard, Localization for precision navigation in agricultural fields—Beyond crop row following, J. Field Robotics 38 (3) (2021) 429–451.
[11] F. Salvetti, S. Angarano, M. Martini, S. Cerrato, M. Chiaberge, Waypoint generation in row-based crops with deep learning and contrastive clustering, in: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part VI, Springer, 2023, pp. 203–218.
[12] Z. Man, J. Yuhan, L. Shichao, C. Ruyue, X. Hongzhen, Z. Zhenqian, Research progress of agricultural machinery navigation technology, Nongye Jixie Xuebao/Trans. Chin. Soc. Agric. Mach. 51 (4) (2020).
[13] M. Martini, S. Cerrato, F. Salvetti, S. Angarano, M. Chiaberge, Position-agnostic autonomous navigation in vineyards with deep reinforcement learning, in: 2022 IEEE 18th International Conference on Automation Science and Engineering, CASE, 2022, pp. 477–484, http://dx.doi.org/10.1109/CASE49997.2022.9926582.

[14] L. Comba, A. Biglia, D.R. Aimonino, P. Barge, C. Tortia, P. Gay, 2D and 3D data fusion for crop monitoring in precision agriculture, in: 2019 IEEE International Workshop on Metrology for Agriculture and Forestry, MetroAgriFor, IEEE, 2019, pp. 62–67.

[15] C.W. Bac, E.J. van Henten, J. Hemming, Y. Edan, Harvesting robots for high-value crops: State-of-the-art review and challenges ahead, J. Field Robotics 31 (6) (2014) 888–911.

[16] R. Berenstein, O.B. Shahar, A. Shapiro, Y. Edan, Grape clusters and foliage detection algorithms for autonomous selective vineyard sprayer, Intell. Serv. Robot. 3 (4) (2010) 233–243.

[17] G. Zhang, T. Xu, Y. Tian, H. Xu, J. Song, Y. Lan, Assessment of rice leaf blast severity using hyperspectral imaging during late vegetative growth, Australas. Plant Pathol. 49 (2020) 571–578.

[18] A. Feng, J. Zhou, E.D. Vories, K.A. Sudduth, M. Zhang, Yield estimation in cotton using UAV-based multi-sensor imagery, Biosyst. Eng. 193 (2020) 101–114.

[19] A. Navone, M. Martini, A. Ostuni, S. Angarano, M. Chiaberge, Autonomous navigation in rows of trees and high crops with deep semantic segmentation, in: 2023 European Conference on Mobile Robots, ECMR, 2023, pp. 1–6, http://dx.doi.org/10.1109/ECMR59166.2023.10256334.

[20] X. Feng, W.J. Liang, H.Z. Chen, X.Y. Liu, F. Yan, Autonomous localization and navigation for agricultural robots in greenhouse, Wirel. Pers. Commun. 131 (3) (2023) 2039–2053.

[21] F. Rovira-Más, I. Chatterjee, V. Sáiz-Rubio, The role of GNSS in the navigation strategies of cost-effective agricultural robots, Comput. Electron. Agric. 112 (2015) 172–183.

[22] R.C. Erenoglu, Reliability of GPS/GNSS-based positioning in a forestry environment, J. Forest. Res. 28 (3) (2017) 605–614.

[23] Y. Yan, B. Zhang, J. Zhou, Y. Zhang, X. Liu, Real-time localization and mapping utilizing multi-sensor fusion and visual–IMU–wheel odometry for agricultural robots in unstructured, dynamic and GPS-denied greenhouse environments, Agronomy 12 (8) (2022) 1740.

[24] M.S.N. Kabir, M.-Z. Song, N.-S. Sung, S.-O. Chung, Y.-J. Kim, N. Noguchi, S.-J. Hong, Performance comparison of single and multi-GNSS receivers under agricultural fields in Korea, Eng. Agric. Environ. Food 9 (1) (2016) 27–35.

[25] S. Zhang, C. Guo, Z. Gao, A. Sugirbay, J. Chen, Y. Chen, Research on 2D laser automatic navigation control for standardized orchard, Appl. Sci. 10 (8) (2020) 2763.

[26] I. Hroob, R. Polvara, S. Molina, G. Cielniak, M. Hanheide, Benchmark of visual and 3D LiDAR SLAM systems in simulation environment for vineyards, in: Towards Autonomous Robotic Systems: 22nd Annual Conference, TAROS 2021, Lincoln, UK, September 8–10, 2021, Proceedings 22, Springer, 2021, pp. 168–177.

[27] A.E.B. Velasquez, V.A.H. Higuti, M.V. Gasparino, A.N. Sivakumar, M. Becker, G. Chowdhary, Multi-sensor fusion based robust row following for compact agricultural robots, 2021, arXiv preprint arXiv:2106.15029.

[28] R. Bertoglio, V. Carini, S. Arrigoni, M. Matteucci, A map-free LiDAR-based system for autonomous navigation in vineyards, in: 2023 European Conference on Mobile Robots, ECMR, IEEE, 2023, pp. 1–6.

[29] V.A. Higuti, A.E. Velasquez, D.V. Magalhaes, M. Becker, G. Chowdhary, Under canopy light detection and ranging-based autonomous navigation, J. Field Robotics 36 (3) (2019) 547–567.

[30] J. Iqbal, R. Xu, S. Sun, C. Li, Simulation of an autonomous mobile robot for LiDAR-based in-field phenotyping and navigation, Robotics 9 (2) (2020) 46.

[31] Y. Bai, B. Zhang, N. Xu, J. Zhou, J. Shi, Z. Diao, Vision-based navigation and guidance for agricultural autonomous vehicles and robots: A review, Comput. Electron. Agric. 205 (2023) 107584.

[32] Y. Zhao, L. Gong, Y. Huang, C. Liu, A review of key techniques of vision-based control for harvesting robot, Comput. Electron. Agric. 127 (2016) 311–323.

[33] P.-c. Huang, Z.-g. Zhang, X.-w. Luo, B.-b. Yue, P.-k. Huang, Monocular visual navigation based on scene model of differential-drive robot in corridor-like orchard environments, Int. Agric. Eng. J. (2019).

[34] J. Kneip, P. Fleischmann, K. Berns, Crop edge detection based on stereo vision, Robot. Auton. Syst. 123 (2020) 103323.

[35] T. Wang, B. Chen, Z. Zhang, H. Li, M. Zhang, Applications of machine vision in agricultural robot navigation: A review, Comput. Electron. Agric. 198 (2022) 107085.

[36] M. Sharifi, X. Chen, A novel vision based row guidance approach for navigation of agricultural mobile robots in orchards, in: 2015 6th International Conference on Automation, Robotics and Applications, ICARA, 2015, pp. 251–255.

[37] M. Zhou, J. Xia, F. Yang, K. Zheng, M. Hu, D. Li, S. Zhang, Design and experiment of visual navigated UGV for orchard based on Hough matrix and RANSAC, Int. J. Agric. Biol. Eng. 14 (6) (2021) 176–184.

[38] J. Chen, H. Qiang, J. Wu, G. Xu, Z. Wang, Navigation path extraction for greenhouse cucumber-picking robots using the prediction-point Hough transform, Comput. Electron. Agric. 180 (2021) 105911.

[39] E. Mendez, J. Piña Camacho, J.A. Escobedo Cabello, A. Gómez-Espinosa, Autonomous navigation and crop row detection in vineyards using machine vision with 2D camera, Automation 4 (4) (2023) 309–326.

[40] J. Radcliffe, J. Cox, D.M. Bulanon, Machine vision for orchard navigation, Comput. Ind. 98 (2018) 165–171.

[41] C. Peng, Z. Fei, S.G. Vougioukas, Depth camera based row-end detection and headland manuvering in orchard navigation without GNSS, in: 2022 30th Mediterranean Conference on Control and Automation, MED, IEEE, 2022, pp. 538–544.

[42] F. Rovira-Más, V. Saiz-Rubio, A. Cuenca-Cuenca, Augmented perception for agricultural robots navigation, IEEE Sens. J. 21 (10) (2020) 11712–11727.

[43] P. Huang, L. Zhu, Z. Zhang, C. Yang, An end-to-end learning-based row-following system for an agricultural robot in structured apple orchards, Math. Probl. Eng. 2021 (2021).

[44] D. Aghi, V. Mazzia, M. Chiaberge, Autonomous navigation in vineyards with deep learning at the edge, in: International Conference on Robotics in Alpe-Adria Danube Region, Springer, 2020, pp. 479–486.

[45] S. Xu, R. Rai, Vision-based autonomous navigation stack for tractors operating in peach orchards, Comput. Electron. Agric. 217 (2024) 108558.

[46] J. Sarmento, A.S. Aguiar, F.N. dos Santos, A.J. Sousa, Robot navigation in vineyards based on the visual vanish point concept, in: 2021 International Symposium of Asian Control Association on Intelligent Robotics and Industrial Automation, IRIA, IEEE, 2021, pp. 406–413.

[47] J. Zhou, S. Geng, Q. Qiu, Y. Shao, M. Zhang, A deep-learning extraction method for orchard visual navigation lines, Agriculture 12 (10) (2022) 1650.

[48] D. Aghi, V. Mazzia, M. Chiaberge, Local motion planner for autonomous navigation in vineyards with a RGB-D camera-based algorithm and deep learning synergy, Machines 8 (2) (2020) 27.

[49] D. Aghi, S. Cerrato, V. Mazzia, M. Chiaberge, Deep semantic segmentation at the edge for autonomous navigation in vineyard rows, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2021, pp. 3421–3428.

[50] Y. Xiao, Z. Lin, Y. Wang, K. Chen, F. Dong, Research on visual navigation technology of citrus orchard based on improved DeepLabv3+ model, in: Proceedings of the International Conference on Computer Vision and Deep Learning, 2024, pp. 1–8.

[51] L. Zhu, W. Deng, Y. Lai, X. Guo, S. Zhang, Research on improved road visual navigation recognition method based on DeepLabV3+ in Pitaya Orchard, Agronomy 14 (6) (2024) 1119.

[52] Z. Yang, L. Ouyang, Z. Zhang, J. Duan, J. Yu, H. Wang, Visual navigation path extraction of orchard hard pavement based on scanning method and neural network, Comput. Electron. Agric. 197 (2022) 106964.

[53] E. Liu, J. Monica, K. Gold, L. Cadle-Davidson, D. Combs, Y. Jiang, Vision-based vineyard navigation solution with automatic annotation, in: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2023, pp. 4234–4241.

[54] M. Martini, A. Eirale, B. Tuberga, M. Ambrosio, A. Ostuni, F. Messina, L. Mazzara, M. Chiaberge, Enhancing navigation benchmarking and perception data generation for row-based crops in simulation, in: Precision Agriculture'23, Wageningen Academic, 2023, pp. 451–457.

[55] M. Martini, M. Ambrosio, A. Navone, B. Tuberga, M. Chiaberge, Enhancing visual autonomous navigation in row-based crops with effective synthetic data generation, Precis. Agric. (2024) 1–22.

[56] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., Searching for mobilenetv3, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1314–1324.

[57] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141, http://dx.doi.org/10.1109/CVPR.2018.00745.

[58] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, IEEE Computer Society, Los Alamitos, CA, USA, 2018, pp. 4510–4520, http://dx.doi.org/10.1109/CVPR.2018.00474, URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00474.

[59] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, 2017, arXiv abs/1706.05587, URL https://api.semanticscholar.org/CorpusID:22655199.

[60] S. Cerrato, V. Mazzia, F. Salvetti, M. Martini, S. Angarano, A. Navone, M. Chiaberge, A deep learning driven algorithmic pipeline for autonomous navigation in row-based crops, IEEE Access (2024).

[61] M. Martini, A. Eirale, S. Cerrato, M. Chiaberge, Pic4rl-gym: a ros2 modular framework for robots autonomous navigation with deep reinforcement learning, in: 2023 3rd International Conference on Computer, Control and Robotics, ICCCR, IEEE, 2023, pp. 198–202.

**Alessandro Navone** achieved a Master's Degree in Mechatronic Engineering in 2022 at Politecnico di Torino, presenting the thesis "Learning Odometric Error in Mobile Robots with Neural Networks". Currently, he is starting his Ph.D. at Politecnico di Torino, in collaboration with the Interdepartmental Centre for Service Robotics (PIC4SeR). His research focuses on AI-driven navigation systems and perception methods for service robotics, with a particular interest in precision agriculture applications.

**Mauro Martini** is a Ph.D. student at Politecnico di Torino. He received from the Politecnico di Torino a Master's Degree with laude in Mechatronic Engineering in 2020, with the thesis "Visual based local motion planner with Deep Reinforcement Learning". He is now carrying out his research activity in collaboration with the Interdepartmental Centre for Service Robotics (PIC4SeR). His research interests currently involve machine learning for autonomous navigation in service and social robotics, with a particular focus on perception and planning and control.

**Marco Ambrosio** obtained a B.Sc. in Mechanical Engineering, in 2018, and a M.Sc. in Mechatronic Engineering, in 2021, both at Politecnico di Torino. His work at PIC4SeR lab as a researcher is focused mainly on UWB localization systems for UGVs. As a mechanical engineer, I am also involved in designing and 3D printing various mechanical components.

**Andrea Ostuni** obtained a B.Sc. in Mechanical Engineering, in 2020 and an M.Sc. in Mechatronic Engineering, in 2022, both at Politecnico di Torino. He is pursuing a Ph.D. at the Interdepartmental Centre for Service Robotics (PIC4SeR), its current research work focuses on localization and perception systems for UGVs. His studies address the problem of autonomous navigation in outdoor and indoor environments.

**Simone Angarano** is a Ph.D. student in Electrical, Electronics, and Communications Engineering and a member of the Interdepartmental Center for Service Robotics at Politecnico di Torino. At the same university, he achieved a Bachelor's Degree in Electronic Engineering in 2018 and a Master's Degree in Mechatronic Engineering in 2020. His research topic is efficient deep learning models for robot perception and control, particularly highlighting key aspects of real-world applications like generalization and robustness.

**Marcello Chiaberge** is currently an Associate Professor within the Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy. He is also the Co-Director of the Mechatronics Lab, Politecnico di Torino, Turin, and the Director and the Principal Investigator of the Interdepartmental Center for Service Robotics (PIC4SeR), Turin. He has authored more than 100 articles accepted in international conferences and journals, and he is the co-author of nine international patents. His research interests include hardware implementation of neural networks and fuzzy systems and the design and implementation of reconfigurable real-time computing architectures.