IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 December 2023, accepted 26 December 2023, date of publication 8 January 2024,
date of current version 12 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3350432

APPLIED RESEARCH

Vid2Cuts: A Framework for Enabling AI-Guided Grapevine Pruning

SIMON HÄRING[1], SOPHIE FOLAWIYO[1], MARIIA PODGUZOVA[1], STEPHAN KRAUß[2], AND DIDIER STRICKER[1,2]

[1]Department of Computer Science, University of Kaiserslautern-Landau (RPTU), 67663 Kaiserslautern, Germany
[2]Department of Augmented Vision, German Research Center for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany

Corresponding author: Simon Häring (simon.haering@dfki.de)

This work was supported in part by the 3rd Call for European Innovation Partnership Agricultural Productivity and Sustainability (EIP-Agri) by the European Agricultural Fund for Rural Development (EAFRD) of the Ministry of Economy, Transport, Agriculture, and Viticulture of Rhineland-Palatinate under Grant 44-10_430 (3. Call_GDV_KI-Rebschnitt); and in part by the Federal Ministry of Education and Research Germany under the Project DECODE under Grant 01IW21001.

ABSTRACT Recent advances in machine learning and computer vision promoted a surge in the development of AI-based approaches aimed at improving various agricultural tasks. In this work, we focus on grapevine pruning, which is one of the labor-intensive tasks in viticulture that requires experienced workers and has a huge impact on grapevine health, future yields and grape quality. Our objective is to develop an AI-based application that provides pruning suggestions according to the "gentle pruning" strategy enabling non-experts in the field to easily engage in the process. To achieve that, we have to deal with multiple challenges such as a large number of grapevine varieties, complicated outdoor conditions characterized by varied light, weather and complex grapevine structures with multiple occlusions. In this work, we present a framework, which allows the generation of pruning suggestions using a video recorded by a smartphone and visualize them in a mobile AR application. Thus, our contributions are the following: 1) we present the collection of a large image segmentation dataset of dormant grapevines; 2) we propose a novel distributed approach to generate pruning suggestions via a semantic 3D grapevine model generated from a smartphone video; 3) we propose a mobile AR application to visualize the pruning suggestions. Results show the robustness of our approach to outdoor conditions as well as reasonable pruning suggestions according to evaluation by domain experts in 71% of cases. We demonstrate the main challenges that must be addressed for such an application and propose a distributed solution to handle them.

INDEX TERMS 3D reconstruction, augmented reality, computer vision, deep learning, grapevine pruning, semantic segmentation.

I. INTRODUCTION

Winter pruning of grapevines is a complicated and time-consuming task. Therefore, many skilled workers are needed to prevent mistakes that may have long-lasting effects on the plants. Since large amounts of skilled workers are hard to find, we seek to provide assistance during the pruning process, opening it up to a larger section of the workforce.

The goal of pruning is to shape the future growth of the plant. This is done by removing unwanted one-year-old branches as well as the spurs and fruiting canes from the previous year. Winemakers can choose between different pruning techniques that optimize for yield, grape quality, resilience to infections or simplicity. In this paper, we build an augmented reality (AR) assistant that enables untrained workers to carry out the method of gentle pruning [1] in particular. The aim of this technique is to reduce the size of the cut wounds, making the plants more resilient to fungal infections.

Computer vision in outdoor environments like vineyards is difficult due to varying lighting conditions, the unique shape of each plant and the similarity between plants in the

The associate editor coordinating the review of this manuscript and approving it for publication was Michele Nappi.

foreground and background. Despite growing in 2D trellises, grapevine canopies exhibit 3D structure and occlusions between branches. We thus construct a 3D model of the plant to which we apply the pruning rules necessary to perform gentle pruning. The 3D information is further used to aid with tracking in the AR application and to correctly position the pruning marks that tell the user where to cut.

Our system is designed to be used by people in the field and aid the pruning process without obstructing it. Therefore, the hardware needs to be small and light-weight which in turn limits computational resources. In order to simplify our current prototypes, we decided to offload the execution of our pipeline to an external desktop computer while the AR application can run on a smartphone.

Our pipeline consists of several consecutive steps. First, a set of diverse keyframes is selected from the monocular RGB input video. Then, semantic segmentation is used to find the constituent parts of the plant and separate them from the background. A point cloud is generated from the same keyframes by a photogrammetry framework and fused with the semantic information in an abstract graph model of the plant. The final pruning suggestions are then determined on this model using a set of rules specific for the gentle pruning method.

We ensure ethical and responsible employment of artificial intelligence in digital agriculture. While AI systems are designed to assist farmers in enhancing production and productivity, they increase risks and the potential for harmful outcomes. Therefore, it is crucial to refer to ethical considerations in our system in order to prevent undesirable consequences. We follow the guidelines provided by [2], which cover a range of ethical considerations essential for developing reliable AI systems for digital farming.

In order to guarantee fairness and eliminate biases, we enhance the diversity of our dataset by collecting a wide range of grapevines with distinct structures from various locations, complemented by employing diverse data augmentation techniques. We implemented a transparent AI system that implies a constant collaboration with farmers and domain experts during the development process, aligning our system closely with their needs and requirements. We provide farmers with comprehensive instructions and detailed descriptions of our system. Moreover, farmers actively participated in the testing and evaluation of our system, which contributed to its improvement. Sustainability is another ethical concept that we consider in our system. In response to feedback from domain experts, we improved our pipeline including the user interface in order to build a user-friendly and trustworthy AI system beneficial to farmers. We ensure the robustness of our AI model, ensuring it performs well across varied environmental conditions. Furthermore, we provide safety by evaluating potential risks and give users the opportunity to decide whether they trust the result based on a confidence metric, which we provide along with the pruning suggestions.

Our contributions can be summarized as follows. We present:

- the collection of a large image segmentation dataset of dormant grapevines,
- a pipeline that extracts 3D and semantic information from a video of a grapevine plant and outputs pruning suggestions using both traditional as well as deep-learning–based methods
- and a mobile AR application to display the results to the user.

After giving an overview about related works in Section II, we describe recording and labeling of the dataset in Sections III-A and III-B, respectively. Our pipeline starts with the extraction of individual frames as explained in Section IV-A. Sections IV-B and IV-C detail the extraction of semantic and 3D information, respectively. These are then combined in a graph as explained in Section IV-D. The process of finding pruning suggestions using this graph is addressed in Section IV-E. Lastly, the AR application and its components are presented throughout Section V.

II. RELATED WORK

Recent years saw a surge in the development of AI-based approaches aimed at improving various agricultural tasks [3]. Machine learning and computer vision algorithms have been widely employed for tasks including yield forecasting [4], [5], fruit detection [6], [7], fruit quality inspection [8], [9], fruit maturity estimation [10], [11], fruit disease diagnosis [12], [13] and tree pruning [14], [15], [16], [17]. In this work, we focus on grapevine pruning.

A. PRUNING SYSTEMS

Grapevine pruning is one of the labor-intensive tasks in viticulture that requires experienced workers and has a huge impact on grapevine health, future yields and grape quality. Developing a fully automated pruning system is challenging due to the complex nature of the grapevine structure, which is associated with a three-dimensional (3D) environment.

Several studies have addressed the problem of grapevine pruning [14], [15], [18], [19], [20], as well as pruning of other fruit trees [21], [22], [23] such as apple trees [16], [24], cherry trees [17], and citrus trees [25]. These studies either focus on the entire pipeline for fully automated pruning [14], [15], [20], [22], [26] or separate steps that are essential for an autonomous pruning system. For instance, machine vision systems have been developed for precise branch detection [16], [17], [18], [24], [25], reconstruction [21], [23], [27], [28], skeletonization [29], [30], [31] and localization of cut positions [32], [33], [34].

You et al. [22] introduced an integrated system for the automatic pruning of sweet cherry trees. The system combines semantic segmentation and a skeletonization algorithm [29] to describe the tree structure and estimate the 3D positions of pruning points. The pruning process is executed by the Universal Robot UR5e. However, human intervention is required to prevent unintentional cuts.

Botterill et al. [14] introduced a robotic system for automated grapevine pruning. It utilizes a mobile platform with stereo cameras to register a grapevine and reconstruct its 3D model. An expert system with a support vector machine (SVM) calculates optimal cut positions, and a robotic arm performs the pruning. While this work is a pioneering effort in automating vine pruning, it requires a significant setup and is time-consuming. Additionally, other plants in the background need to be blocked using a blue blanket.

In their work, Fourie et al. [15] proposed a system that utilizes a recurrent graph neural network (GNN) [35] to directly generate pruning rules from grapevine structures. This approach offers an objective method for constructing a pruning scheme, minimizing potential discrepancies that can occur when experts rely on their individual expertise. However, this system has only been trained and tested on a limited synthetic dataset, therefore, relies on multiple assumptions and simplifications. The adaptation of the system to real data and the testing of its robustness remain open issues.

Gentilhomme et al. [18] developed ViNet, a deep learning approach that accurately understands the grapevine structure. By incorporating semantic segmentation and graph generation, ViNet detects nodes and branches from 2D images. However, the authors used an artificial background and relied on a dataset primarily consisting of clear grapevine structures without occlusions or atypical branch intersections, which are commonly encountered in natural environments.

The current state of the art as described above proves the general feasibility of pruning suggestions systems but does not address the problems of real-world application such as challenging outdoor conditions, complex plant structures with significant occlusions and unusual shapes and branching. In this work, we present a distributed system that includes a complete pipeline for the generation of grapevine pruning suggestions using video recordings captured by a smartphone camera. Our system eliminates the need for complex setups involving artificial backgrounds, simplified environments, robots, or specialized cameras. Instead, it relies on the segmentation of a reconstructed 3D model to achieve accurate localization of grapevine parts and calculate the optimal pruning positions. The results are conveniently visualized through a mobile application, enabling non-experts to easily engage in grapevine pruning.

B. PLANT RECONSTRUCTION

In this study, we tackle the challenges presented by complex outdoor conditions, specifically focusing on grapevines with unclear structures and numerous occlusions. Due to the limitations of 2D images in accurately assessing branch connections, we adopt a 3D reconstruction approach to effectively localize relevant parts of the grapevines.

Various research efforts have been dedicated to addressing the reconstruction of trees and plants, which can be broadly categorized into active and passive approaches [36], [37], [38]. Active methods involve specialized equipment and rely on external light sources with known parameters and locations [38], [39], [40]. Whereas passive methods extract depth information directly from single or multiple images to reconstruct a 3D scene [37], [38]. Passive approaches are more flexible and cost-effective [41]. Several studies [37], [41] have concluded that the passive photogrammetry approach combining Structure-from-Motion (SfM) [42] and Multi-View Stereo (MVS) [43] provides high flexibility and stability in outdoor conditions. Consequently, they consider this approach to be the more suitable method for modeling plants.

At present, there are various open-source computer vision frameworks available for photogrammetry, such as COLMAP [44], [45], Meshroom [46], BoofCV,[1] as well as commercial software like Metashape[2] and RealityCapture.[3]

In line with this, we employ the photogrammetry approach for the reconstruction of grapevines, taking advantage of its flexibility and stability in capturing accurate plant structures outdoors. To meet our specific requirements, we customized Meshroom, a photogrammetry software, which demonstrated high robustness and stability in outdoor environments (see Section IV).

C. PLANT SEGMENTATION

Datasets and methods for leaf segmentation [47], disease detection [48], [49] and fruit detection [50], [51], [52] are numerous. Whereas such tasks involve the handling of occlusions from leaves, dormant plants in winter pose different challenges like thin structures spanning across large parts of the image. Existing datasets for semantic segmentation of dormant grapevines lack either in size or in the variability of viewpoints and backgrounds [18], [53], [54], [55].

Besides varying outdoor lighting conditions that most computer vision methods in agriculture have to face, an additional difficulty in vineyard and orchard environments is the similarity between plants in the foreground and background. Previous methods chose artificial backgrounds [14], [18] or depth-based background removal to alleviate this. Specifically, Majeed et al. [56] performed semantic segmentation on young, dormant apple trees in an orchard. They trained their network once with the original RGB images and once with images from which they removed the background using depth information acquired using an RGB-D camera. Doing so, they found that the foreground-only network performed better.

Borrenpohl and Karkee [57] separated foreground and background in images of dormant sweet cherry trees using depth information. They did this in order to swap the background of one sample with the background of another sample and thus increase the effective size of their dataset. We employ a similar data augmentation technique using the ground truth to extract the foreground. Using this dataset,

[1] BoofCV: http://boofcv.org/
[2] Agisoft Metashape: http://www.agisoft.com
[3] Epic Games RealityCapture: https://www.capturingreality.com/

they trained two instance segmentation networks to find the vertical fruiting branches of sweet cherry trees. One of the networks was trained with naturally lit images and the other one with artificially lit images. Borrenpohl and Karkee found the network trained on artificially lit images to achieve better results.

D. MOBILE AUGMENTED REALITY

The interest in using AR to assist different agricultural tasks is growing [58], [59], [60], [61] but applications are still limited. An overview of use cases and methods regarding precision farming is given in [62].

Besides those, there are several monocular 3D reconstruction and depth estimation frameworks available which enable realistic AR effects on the mobile phone for general use. Yang et al. [63], for instance, generate dense geometrical structures from a monocular smartphone camera in real time. Furthermore, researchers in [64] use the RGB camera of a smartphone to estimate dense depth maps allowing them to render virtual objects in real time. Both frameworks [63], [64] are based on either a Visual Inertial Odometry (VIO) or a Simultaneous Localization and Mapping (SLAM) system that tracks six degree of freedom (6DoF) poses for selected keyframes.

There is an increasing number of vision-based monocular SLAM and odometry systems with proven real-time performance for mobile applications, either using a single camera or fusing RGB data and data from inertial sensors. ORB-SLAM [65] and VINS-Mono [66] are examples for feature-based methods whereas LSD-SLAM [67] is based on direct image alignment.

AR applications for agricultural environment are still limited while other existing mobile AR applications are commonly applied on indoor scenes where they reconstruct large surfaces with little detail. However, visualizing virtual content on grapevines involves the handling of challenging weather conditions, including severe illumination changes, as well as complex grapevine structures that contain overlapping and thin branches. We therefore opted to develop a customized mobile tracking solution which addresses these challenges.

III. DATASET

We collected a large image segmentation dataset of dormant grapevines for winter pruning. The dataset contains over 11k labeled images from 2.5k videos of the plants as they appear in the field with realistic background. Each sample includes two semantic masks. The first mask contains the stem, two-year-old branches, one-year-old branches, nodes and trellis wires. The second one highlights special areas of the plant: cut wounds, dried-up branch segments and buds.

A. IMAGE ACQUISITION

Data was collected over three consecutive winters starting in 2020/2021 in six vineyards located in wine-growing regions at the river Moselle and in Rhine Hesse in Germany. The dataset covers multiple grape varieties, locations as well as weather conditions and was recorded using multiple cameras and movement patterns.

Initial videos were taken with the Intel RealSense L515 and D435i cameras. In the following winter the integrated RGB camera of the XREAL light AR glasses was used and in the last year a switch to the camera of the Samsung Galaxy S20 FE 5G phone was made, due to its higher image quality. Resolutions range from 640×480 to 1920×1080.

Two different recording patterns were applied to cover one side of the plant from multiple different angles. First, we used a clockwise motion starting in the top left, while always keeping the camera pointed towards the top of the stem. Later, a zigzag pattern described in Section IV-A was adopted as it allowed for more diverse view points.

B. LABELING TOOL

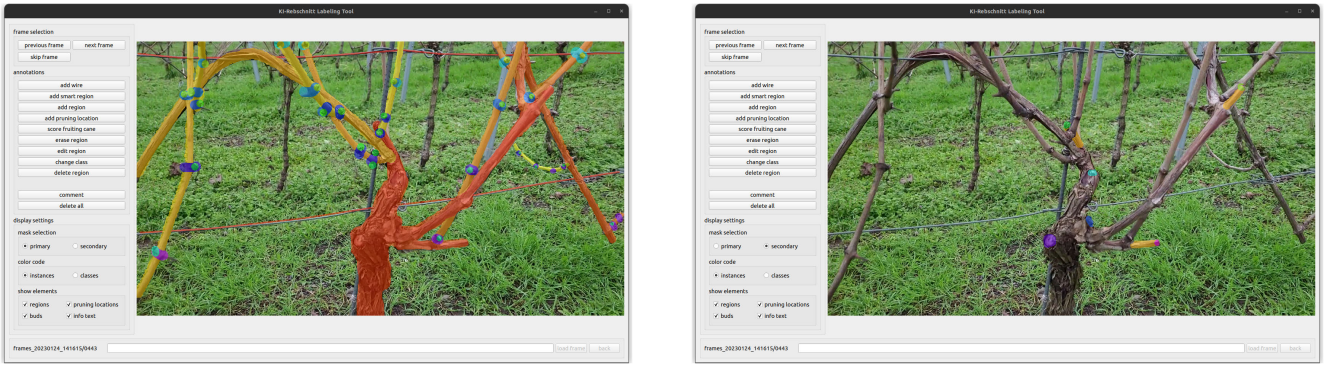
The input to our labeling tool are videos. We use the selection process described in Section IV-A to extract up to ten frames from different viewpoints for each video. Those are then labeled for instance segmentation. An example of our tool can be seen in Fig. 1.

We built a labeling tool that reduces the effort of annotating the types of images contained in our dataset. The relevant structures in images of grapevine plants are often long and thin (e.g., wires, branches) which is not ideal for hand-drawn regions. Therefore, our tool supports three ways of defining regions:

1) *wire region*: a line strip of constant width,
2) *smart region*: defined as a line strip with variable width but with its final shape determined via GrabCut [68],
3) *region*: drawn by the user using a circular brush of variable diameter.

The trellis wires are marked using line strips, since a single line cannot account for bends that appear near attachment posts or in places where it supports last year's fruiting cane. We use what we call smart regions for one year and two year old wood. They are based on the idea that the branches of a plant are line strips of variable thickness. The user creates a line strip by clicking along the branch with a rough estimate of the local thickness. As seen in Fig. 2, once $n > 1$ points are clicked, a region is estimated using the GrabCut [68] algorithm and added to the region of the current instance. p_{n-1} and p_n define the *foreground* and *possibly foreground* masks needed by GrabCut. Everything else is considered background. This can produce detailed masks requiring only a low number of clicks by the user. However, in situations of low contrast between foreground and background (e.g., brown dirt) or high contrast on the branch (e.g., from hard shadows) the quality of the region suffers. The masks for the stem, nodes, buds and health factors are painted with a circular brush of variable size.

For each instance we store its mask, its class as well as its underlying line strip if available. Since we want to train a network for semantic segmentation, we convert the collected information into the two class masks described at the start



(a) Primary instance mask (b) Secondary instance mask

FIGURE 1. Primary and secondary masks for one frame viewed in our labeling tool. The primary mask covers the entire foreground plant and splits it based on wood age. It also contains nodes and buds. The second mask focuses on cut wounds and dry spots.

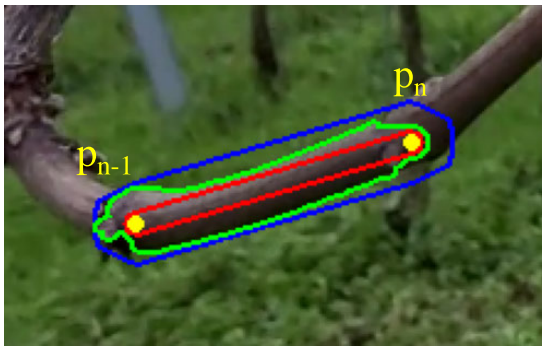


FIGURE 2. Two clicks by the user define points p_{n-1} and p_n which in turn define the *foreground* (red outline) and *possibly foreground* (blue outline) regions as lines of different thickness. The green outline is the resulting *foreground* region after GrabCut [68] is run.

of Section III. As we store the user-generated line strips for the two branch classes, as well as the order in which the object instances were labeled, we can later infer additional information. This includes regions of occlusions, 2D growth directions of branches, and a relative depth-wise ordering of the branches.

IV. FRAMEWORK OVERVIEW

In this section we describe the framework, which allows generating pruning suggestions for grapevine pruning using a video from a smartphone as input. The pipeline, illustrated in Fig. 3, includes the following stages.

1. Video Capture & Keyframe Selection: A grapevine is recorded by a smartphone camera, ensuring that the captured video contains all necessary details about the plant. Then, the optimal keyframes are selected such that they allow focusing on significant information and enable reconstructing the point cloud as accurately as possible (Section IV-A).

2. Semantic Segmentation: Semantic segmentation of keyframes identifies the constituent parts of a grapevine and extracts information, which is essential for pruning. Moreover, background estimation is applied such that only foreground information is used for the reconstruction stage (Section IV-B).

3. Point Cloud Generation: The selected keyframes are used for the generation of a dense point cloud of grapevines employing Meshroom [46]. The 2D semantic information extracted earlier is combined with the generated point cloud to construct a 3D segmented model of the grapevine (Section IV-C).

4. 3D Graph Generation: A 3D graph is constructed based on the segmented grapevine model generated in the previous stage (Section IV-D).

5. Pruning Suggestions: Pruning suggestions are computed by evaluating the properties of the reconstructed grapevine using the 3D graph (Section IV-E).

6. Mobile AR Application: Finally, the results are visualized in the mobile AR application, which includes tracking grapevine parts (Section V-A) and visualization of pruning suggestions (Section V-B).

In the following, we provide detailed information about each stage of the pipeline.

A. KEYFRAME SELECTION

The primary objective of this work is to precisely estimate cutting positions for grapevines using a monocular video sequence and generate suggestions for gentle pruning. To achieve high localization accuracy, we employ a 3D reconstruction technique to model individual plants and obtain spatial semantic information.

The first challenge that needs to be solved is the proper selection of the most representative keyframes from the video sequence. This crucial step involves striking a balance between performance and quality. The number of selected keyframes significantly impacts the overall system performance, while the quality of these keyframes directly influences the accuracy and reliability of the resulting 3D model, thereby affecting the overall outcome.

The selected keyframes should satisfy the following properties:

1) *Sufficient Feature Sharing*: They should share a reasonable number of features that are suitable for the subsequent 3D reconstruction process.

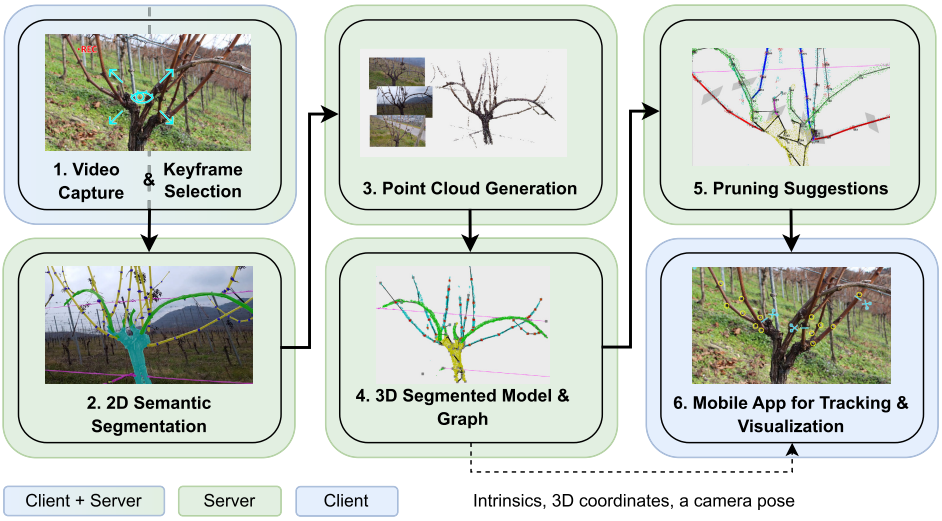


FIGURE 3. The pipeline of the distributed approach for generating grapevine pruning suggestions. It consists of several consecutive stages described in the following sections. 1. Video capture (a client) and keyframe selection (the server) (Section IV-A). 2. Semantic segmentation (Section IV-B). 3. Point cloud generation (Section IV-C). 4. 3D graph generation (Section IV-D). 5. Pruning suggestions (Section IV-E). 6. Mobile AR application for tracking (Section V-A) and visualization (Section V-B). The reconstructed point cloud with intrinsic parameters and estimated 3D poses are utilized for the generation of a 3D segmented grapevine model, which is further employed in the mobile AR application.

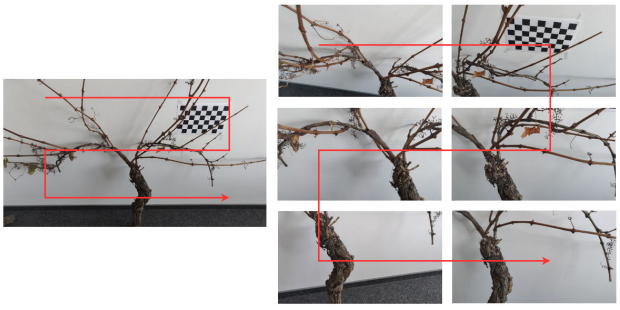


FIGURE 4. An example of the zigzag pattern applied for indoor plant recording. The pattern involves capturing videos of the grapevine by alternating between moving from right to left and vice versa, creating a zigzag shape. This approach ensures comprehensive coverage of the plant from different angles and perspectives, allowing for a thorough registration of its structure and details.

2) *Diversity*: They should be diverse in terms of content and viewpoints. This diversity helps to capture different aspects of the grapevine structure.
3) *Representativity*: They should represent the entire video sequence, ensuring that the subset of frames selected is sufficient to capture all significant details.
4) *Minimal Noise and Blur*: They should exhibit minimal noise and blur to ensure the highest possible quality of the 3D model as well as the final results.

For property 1, we employ a zigzag pattern during the recording process illustrated in Fig. 4. The pattern involves capturing videos of the grapevine by alternating between left-to-right and right-to-left motion, creating a zigzag pattern. This approach allows to capture the most crucial parts of the plant from various viewing angles, while ensuring an adequate amount of matching information between frames required for 3D reconstruction.

For property 2, we divide the video into N sectors of equal size, where $N = 4$ is used in the experiments. The sector size is determined based on the total number of video frames. These sectors correspond to the most distinctive regions based on the employed recording pattern. This approach guarantees that within each sector, the keyframes selected include a high number of shared features.

For property 3, we leverage semantic information to identify keyframes, which contain a substantial amount of foreground information. To ensure the representativity of the selected keyframes, we filter out frames that are less informative and include less than 30% of foreground pixels, while also excluding frames that do not contain any pixels classified as stem. This filtering process is particularly relevant for the frames in the initial and final sectors, which correspond to the beginning and end of the recording. This step effectively reduces the number of selected keyframes while enhancing the significance of the most informative ones.

For property 4, we utilize the variance of the Laplacian as a measure of image blur for frames within each sector. The Laplacian operator, a second-order derivative commonly used for edge detection in computer vision, provides insights into the presence and sharpness of edges in an image [69]. By estimating the variance of the Laplacian, we can infer the spread of edge responses. Consequently, a lower estimated variance indicates a reduced presence of edges, suggesting that the image is more likely to be blurred [70].

Our goal is to select keyframes with the highest variance such that there are uniformly distributed within each sector. To this end, we use an adaptive variance threshold with

an initial mean value and adaptive window size per sector. Specifically, we aim to select a varying number of frames, denoted as M_i, within the range of $min\_n_i \leq M_i \leq max\_n_i$, where $i = \{1..N\}$, in order to prevent the dominance of sectors with low mean variance. Frames that do not meet the variance threshold are filtered out during the selection process. By employing adaptive thresholds, we ensure a uniform distribution of frames and maintain representativity within each sector. This allows us to capture necessary details and preserve variations across the entire video while avoiding an excessive concentration of frames from any specific sector.

B. SEMANTIC SEGMENTATION

Semantic segmentation is used to separate the foreground from the often similar background before 3D reconstruction and to find certain parts including nodes and branch types which are important for the later steps in our pipeline. Fig. 5 shows an example of our semantic segmentation. Classes of the primary mask are shown in solid colors: stem (yellow), two-year-old wood (green), one-year-old wood (cyan), nodes (red) and wires (pink). The secondary classes: cut wounds, dry areas and buds are visualized by purple, gray and blue borders, respectively. Note the challenges posed by the similarity between the stem in the foreground and the soil in the background as well as the dissimilarity along the two year old branch in the bottom left due to the hard shadow.

1) ARCHITECTURE

We use a modified version of the Deep Dual-Resolution Network (DDRNet) described by Pan et al. [71] to segment a selection of frames into a hierarchy of foreground and background classes. DDRNet consists of a low resolution branch with a pooling module at the end to capture high-level information across large parts of the image and a more shallow high-resolution branch that can maintain details of small objects like buds. We further augment this by keeping a set of early, high resolution feature maps that are concatenated after the fusion of the previously mentioned branches. Fig. 6 shows the structure of our network with feature map resolutions given relative to the input resolution.

2) TRAINING

During training we use a combination of dice loss and automated focal loss [72]. The original focal loss [73] shifts the influence on the loss towards hard samples to counteract class imbalance. This can lead to diminishing gradients when fewer samples are considered hard as the network improves. Automated focal loss seeks to remedy this, by relaxing the focus as the training progresses. Like [71] we calculate the loss for the final output as well as an auxiliary output. The latter is derived from intermediate features captured after the first fusion of the high and low resolution branches (after the the leftmost green block in Fig. 6). This output is generated by a separate segmentation head that is inactive during inference.

3) DATA AUGMENTATION

An outdoor system in the domain of agriculture is required to handle a variety of lighting conditions and plant shapes. Therefore, we employ a wide range of data augmentation techniques such as random cropping, horizontal flipping, rotation, adding noise and motion blur as well as shifting brightness, color and contrast. In addition to those, we randomly add artificial snow [74], glare and foreign objects including hands from [75]. To achieve better independence from the background, we occasionally replace it with images from [76] or the background of other samples from our dataset similar to [57]. The latter is done by transplanting background patches on top of the foreground region of one sample in order to create a background-only image which then receives the foreground region and class labels of another sample. An example of this can be seen in Fig. 7.

C. POINT CLOUD GENERATION

Precise localization of grapevine parts is crucial for generating accurate cutting suggestions in order to achieve careful pruning. Relying exclusively on 2D information is not reliable due to the inability of images to accurately represent the complex spatial structures of real grapevines and identify their interconnections. To overcome this limitation, we generate a dense point cloud of grapevines using Meshroom, a photogrammetric computer vision framework developed by AliceVision [46].

1) DENSE POINT CLOUD

Photogrammetry aims to reconstruct 3D scenes from a set of unordered images. It addresses the challenge of regaining depth information, which is lost when transforming the 3D world into a 2D image. The generation of the point cloud combines the results of two computer vision algorithms: Structure-from-Motion (SfM) [42] and Multi-View Stereo (MVS) [77].

The accuracy of the point cloud is heavily influenced by the quality of the images used for reconstruction. To enhance the point cloud quality, we performed additional processing on the keyframes selected with the procedure described in Section IV-A. This involved extracting essential grapevine-related information while excluding irrelevant data that is typically included in complex outdoor scenes, such as background elements, nearby plants and other extraneous objects. To mitigate the influence of this irrelevant information, we applied semantic segmentation to remove the background from the selected keyframes. This approach enables us to accelerate the generation process and obtain a more informative point cloud by removing irrelevant elements and focusing on essential plant-related information.

For the feature extraction step, we apply a combination of DSP-SIFT [78] as well as AKAZE [79] features. Our approach involves utilizing a set of 2000 features to achieve a trade-off between system performance and the quality of the generated point cloud.

(a) Original image. (b) Segmented image.

FIGURE 5. An example of our semantic segmentation (best viewed in color). The main classes are stem (yellow), two-year-old wood (green), one-year-old wood (cyan), nodes (red) and wires (pink). Cut wounds (purple outline), dry areas (gray outline) as well as many buds (blue outline) were not detected in this case.

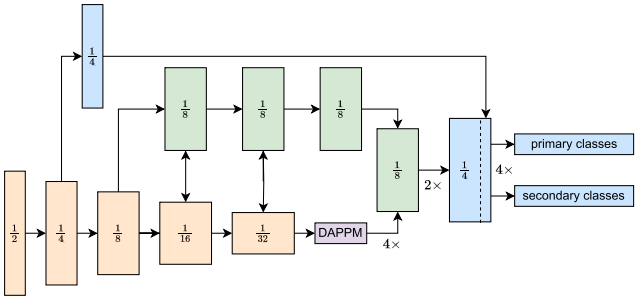


FIGURE 6. The structure of DDRNet [71] with our modifications shown in blue. A stack of early features has its channel dimension reduced (top) before being concatenated to the sum of the high and low resolution branches (right). The two sets of class probabilities are estimated via two separate soft-max activations.

We apply Semi-Global Matching [80] to estimate the disparity between image pairs, which provides pixel-wise depth information. The computed depth maps are then used to build an optimal dense point cloud [81], [82].

2) SEGMENTED POINT CLOUD

In order to obtain a segmented 3D model of a grapevine, we combine the previously extracted 2D semantic information with the generated point cloud. This process utilizes the necessary camera parameters and poses retrieved during the point cloud generation stage. The following procedure is employed:

1) First, we apply the algorithm introduced by Katz et al. [83] to identify the visible points in the point cloud for each viewpoint. These points represent the grapevine parts that are observable from that particular perspective.
2) Given intrinsic and extrinsic camera parameters, we compute projections of 3D coordinates $\mathbf{X}$ to their corresponding 2D views and establish correspondences between the 3D coordinates and the projected 2D pixels, denoted by $\mathbf{x}$. The projection is given by:

$$\mathbf{x} = \mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{X}, \qquad \mathbf{t} = -\mathbf{R}\mathbf{c}, \tag{1}$$

where $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the intrinsic matrix, $\mathbf{R} \in \mathbb{R}^{3\times3}$ is the rotation, $\mathbf{t} \in \mathbb{R}^3$ is the translation, $\mathbf{c} \in \mathbb{R}^3$ is the camera center in world coordinates.
3) Next, we extract the semantic class for each projected pixel. This allows us to assign a semantic label to each point in the point cloud.
4) By collecting the semantic classes from all keyframes, we determine the most commonly predicted class for each 3D point.

Fig. 8 shows the dense point cloud generated from a set of keyframes, along with the corresponding segmented point cloud that incorporates the classes described in Section IV-B.

D. 3D GRAPH GENERATION

We collect all the information needed to decide which branches to keep and where to cut in 3D space in a tree graph representing the plant. The vertices of the graph correspond to 3D points on the grapevine, while edges represent the branch segments between two neighboring points. As each vertex has a 3D position, the resulting graph is a skeleton of the plant. Fig. 8b shows an example of a 3D graph with its corresponding segmented point cloud.

1) SCALE

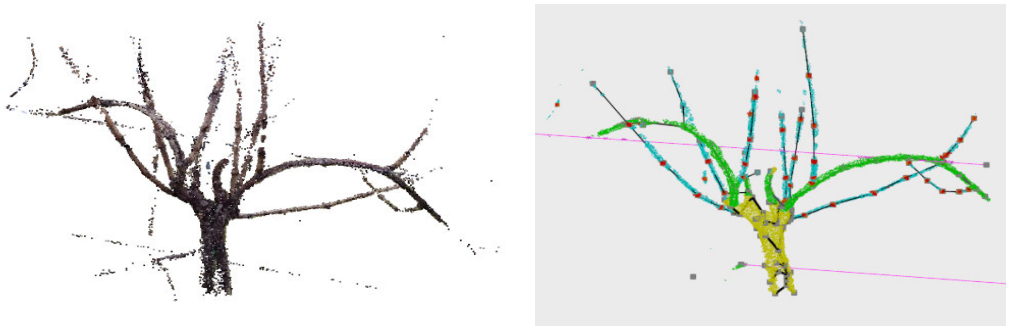
Multiple steps of the graph generation depend on distances. Therefore, we need a sense of scale which SfM cannot recover. Thus, we measure the distance between the lower two wires and compare it to a user-supplied value for the respective vineyard. This is done by first separating the wire point cloud into clusters and then fitting a 3D line into each cluster. The clusters are created by projecting the wire points into a frame where both wires are visible and then assigning each point to the closest of the two. Any remaining outliers due to issues in the generation or segmentation of the point cloud are discarded by RANSAC [84] which is used to fit 3D lines into the clusters.

2) PLANT STRUCTURE

Once a scale is established, graph vertices are uniformly distributed across the point clouds in a way that all of the

(a) Background donor. (b) Background of (**a**). (c) (**b**) and (**d**) combined. (d) Foreground donor.

FIGURE 7. Image (a) has its foreground replaced by background patches, giving image (b). This is then used to replace the background of another image (d). The result is shown in (c).



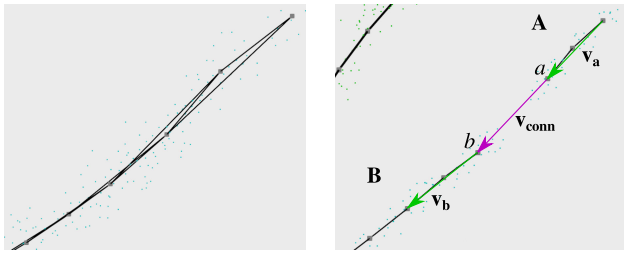
(a) Dense point cloud. (b) Segmented point cloud and 3D graph.

FIGURE 8. An example of a segmented dense point cloud. (a): a dense point cloud built with Meshroom [46]. (b): the corresponding segmented point cloud with the classes described in Section IV-B and the resulting 3D graph.

plant's nodes are included. The lowest vertex on the stem will later become the root of the directed tree. Additional information like thickness, cut wounds and the position of buds relative to their respective nodes are attached to nearby vertices. Positions for nodes, buds and cut wounds are given by clustering their respective segmented point cloud and recording the cluster centers. The thickness is estimated by measuring the diameter of the point cloud at multiple positions along a branch.

Next, the vertices within a class are connected to each other (e.g., only those on one-year-old wood) using distance thresholds. Further, an angle-based criterion is used to prevent triangles along the branches in places of higher vertex density, as shown in Fig. 9a. This is done by only allowing vertices a and b to connect if neither of them has any previous edges within a cylinder around their connecting vector $b - a$. These initial edges result in a set of connected components in the graph. Each connected component can either be a part of a branch, a complete branch or contain parts of multiple branches.

In order to keep branch ages consistent, we can bridge a connected component either to one of the same age or to an older one that is closer to the stem. Both cases have different connection criteria: connections between branch segments of the same age need stricter angle thresholds than those between segments of different ages, since a branch keeps growing in approximately the same direction, while an offshoot grows laterally. As shown in Fig. 9b, when trying to connect branch segments **A** and **B** at vertices a and b the local direction of a branch segment is defined as the vector between the vertex that takes part in the connection



(a) Sharp triangles due to wrong connections. (b) Vectors for angle calculation between segments **A** and **B**.

FIGURE 9. Close-ups of different scenarios observed during graph generation. (a): the result of purely distance based connections between vertices. (b): vectors needed to calculate the distance and angles between branch segments.

and one of its neighbors. Angles are then calculated between $\mathbf{v_a}$ and $\mathbf{v_{conn}}$ as well as $\mathbf{v_a}$ and $\mathbf{v_b}$ using the dot product. Two vertices a and b are connected if the distance between them, the aforementioned angles and a set of weights for the different combinations of wood types are within acceptable ranges. These weights and thresholds were determined experimentally. We found dynamic angle thresholds based on the distance between a and b to work well.

3) POSTPROCESSING

After connections between branch segments are made, the minimum spanning tree of the largest connected component in the graph is calculated. Its edges are oriented such that they point away from the root vertex located at the bottom of the stem. Further filtering is done to split cases of crossing branches whose point clouds have merged leading

Algorithm 1 Split Crossing Branches

Require: connected component with ≥ 4 degree-1 vertices
  $h_{search\_dist} \leftarrow 3$
  $y_{search\_dist} \leftarrow 7$
  $v \leftarrow$ first degree-4 vertex
  if v exists **then** ▷ X-type crossing
    find best-aligned pair of incident edges
    remove all other edges of v
    connect former neighbors of v to each-other
  else
    $v_1 \leftarrow$ first vertex with degree-3
    $\alpha_1, \alpha_2, \alpha_3 \leftarrow$ pairwise angles between incident edges
    if $\alpha_i \approx \pi \wedge \alpha_{j\neq i} \approx \frac{\pi}{2}$ **then** ▷ H-type crossing
      look along the edge corresponding to $\alpha_{j\neq i}$ for another degree-3 vertex v_2 at most h_{search_dist} steps away
      if found **then**
        remove all edges between v_1 and v_2
      end if
    else
      if one $\alpha_i < \frac{\pi}{2}$ **then** ▷ Y-type crossing
        look along the edge corresponding to $\alpha_{j\neq i}$ for another degree-3 vertex v_2 at most y_{search_dist} steps away
        if found **then**
          find the edge incident to v_2 that aligns best with the edge corresponding to α_i at v_1
          remove those edges and connect their former incident nodes directly
        end if
      end if
    end if
  end if

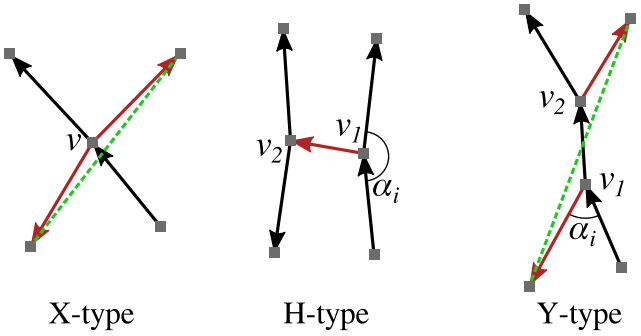


FIGURE 10. Three different types of crossing branches that have merged in the point cloud and graph. Arrows are edges of the graph in the direction that they are traversed in algorithm 1. Removed and added edges are shown as red arrows and green dashed lines respectively.

to unwanted edges. This step is detailed in algorithm 1 and Fig. 10.

Finally, the graph is simplified by removing most vertices that do not represent nodes, cut wounds, junctions or end points of branches. Some non-essential vertices are kept such that the edges of the graph adhere more closely to the actual shape of the plant.

E. PRUNING SUGGESTIONS

The objective of this study is to provide rule-based pruning suggestions according to the "gentle pruning" strategy. It refers to a pruning technique used to maintain grapevine health and productivity while minimizing potential damage to the plant [1].

For the generation of pruning suggestions, we leverage both 2D and 3D information obtained from the preceding stages in a ranking-based recommendation system that we created. This system assesses the characteristics of grapevine branches to identify the optimal candidates for fruiting canes, which will yield the highest fruit production in the present year, as well as new spurs for future harvests. The entire process of generating these suggestions involves three key steps: 1) evaluation of grapevine properties and ranking of branches, 2) grapevine traversal and decision making and 3) generation of cutting suggestions.

Fig. 11 illustrates a keyframe of the grapevine and the corresponding segmented 3D graph with the generated pruning suggestions. The example depicts the optimal scenario where two future spurs and two fruiting canes (one for each side of the plant) can be identified.

1) STEP 1: GRAPEVINE PROPERTIES EVALUATION & RANKING

In order to assess the suitability for pruning decisions, we introduce a ranking system that takes into account the characteristics of branches essential for pruning. This system incorporates both rewards and penalties for branches.

In terms of rewards, higher scores are assigned to branches that possess desirable characteristics such as being one year old, having sufficient length and a number of nodes with appropriate directions of detected buds and an adequate thickness. Additionally, the ranking score takes into account the origin type, branch position relative to the root node, the estimated up direction and the existing wires. These attributes significantly contribute to determining the optimal pruning strategies for grapevines.

Conversely, branches with unfavorable traits, including being two years old or a stem, broken or dry, excessively short or too distant from the stem, are assigned lower scores as part of the punitive side of the ranking system. In the decision-making process these branches are considered to be less suitable.

In the ideal scenario, our objective is to identify two fruiting canes and two future spurs, i.e. the essential branches. To achieve this, we divide the plant into two sides (**A** and **B**) and conduct ranking separately for each side.

2) STEP 2: GRAPEVINE TRAVERSAL & DECISION MAKING

The objective of this step is to analyze the evaluated branches and identify the most suitable candidates for fruiting canes and future spurs. During this process, we consider various potential scenarios, including the following cases.

(a) An example of selected keyframes.

(b) The segmented point cloud and computed pruning suggestions.

FIGURE 11. An example of the segmented 3D graph and the generated pruning suggestion. (a): a keyframe with the original grapevine. (b): the corresponding processed segmented point cloud. The red branches represent potential candidates for future spurs. The blue and dark blue branches indicate the top two candidates for a fruiting cane, i.e. priority 1 and priority 2, respectively. The gray planes mark the designated cut positions. This example demonstrates an optimal scenario where two future spurs and two fruiting canes (one for each side of the plant) can be identified.

Scenario 1: Two Fruiting Canes & Two Future Spurs: In the best-case scenario, where healthy grapevines have been consistently pruned using gentle pruning in previous years, it is possible to identify a fruiting cane and a future spur for both plant sides. However, our system is designed to be applicable to various grapevine types, including those previously pruned using different techniques. Therefore, we also consider more challenging scenarios.

Scenario 2: One vs Two Fruiting Cane(s): In this particular scenario, we encounter situations where there is an inadequate number of branches available for one or both sides of the plant. This poses a challenge in determining the essential branches, and the system must decide in favor of one or two fruiting canes. If candidates for fruiting canes are found for side **A** and not found for side **B**, we assess the feasibility of selecting the top two candidates on side **A**. If achieving this becomes unfeasible, e.g. few branches on side **A**, we keep the option of having only one fruiting cane as the final solution.

Scenario 3: One Fruiting Cane & One Future Spur: In the scenario of one-sided plants, where there are no branches available on one side, we have two possibilities. If there is a sufficient number of candidates, we aim to identify two future spurs and one fruiting cane and follow Scenario 2. Alternatively, if the number of suitable candidates is limited, we select one fruiting cane and one future spur as the optimal choice.

Scenario 4: Fruiting Cane vs Future SpurL In certain scenarios, the same branch might receive the highest rank as both a fruiting cane and a future spur. To address this, we conduct a final verification by scoring the branch using subsets of properties that are utilized in the ranking process during Step 1 (Section IV-E1) and specifically refer to either a fruiting cane or a future spur. This additional evaluation helps to ensure that the selected branches for fruiting canes and future spurs are distinct and fulfill the necessary requirements.

Through this decision-making process, we identify the branches that have the highest potential to serve as future spurs and fruiting canes. Subsequently, we utilize this information to generate pruning suggestions that are tailored to the specific needs of grapevines.

3) STEP 3: CUTTING SUGGESTIONS GENERATION

As a final step, we leverage the decision-making outcomes from the previous stage and generate the pruning suggestions using the cutting rules summarized in Table 1.

a: FUTURE SPURS

To determine the cutting position for a future spur, we assess the orientation of the identified buds. Our objective is to detect the edge where the source node exhibits a visible bud pointing towards the ground or sideways, while the target node displays a visible bud pointing towards the sky or sideways. The subsequent edge following the second suitable node is designated as the cutting edge, with a cutting distance equivalent to half of the edge length.

b: OTHER BRANCHES

To address the remaining branches, we generate cutting positions for the edges where the source node corresponds to a basal node of the current branch. We define the following cutting rules:

- *Other one-year-old branches.* Cut with a minimum possible distance equal to a small predefined epsilon value.
- *Two-year-old branches.* Cut with a distance equal to the average between the minimum and maximum possible distances. The minimum distance is determined by the thickness of the branch, while the maximum distance is defined as below the next node.
- *Dry branches.* Cut without any distance specified.

Furthermore, we perform an additional check to prevent conflicting cut positions, aiming to visualize the earliest

TABLE 1. Cutting rules based on the "gentle pruning" methodology. Firstly, we determine the optimal cutting edge, considering the target and branch type. For future spurs, we identify an edge following the second suitable node, where the source node has a visible bud pointing downwards or sideways (↓ / →), and the target node has a visible bud pointing upwards or sideways (↑ / →). For other branches requiring cutting, the edge after the first suitable node with a basal bud is given priority. Secondly, once the suitable edge is identified, we generate the precise cutting position for this particular edge.

Branch Type	Nodes	Position on Branch	Position on Edge
Future spur	1^{st} node: bud ↓/→ 2^{nd} node: bud ↑/→	after 2^{nd} suitable node	middle of edge
Other 1-year old	1^{st} node: basal bud	after 1^{st} suitable node	eps distance
2-year old	1^{st} node: basal bud	after 1^{st} suitable node	$(min_d + max_d)/2$
Dry branches	1^{st} node: basal bud	after 1^{st} suitable node	without distance

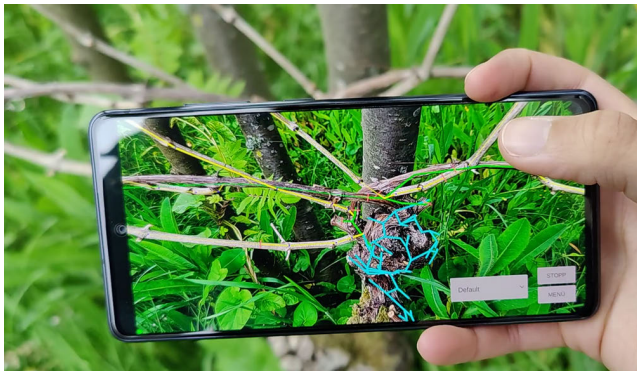


FIGURE 12. Mobile AR app running on a smartphone.

feasible position for each branch while minimizing the number of cuts.

V. MOBILE AR APPLICATION

We build a mobile augmented reality application to visualize cut positions in the field after processing a single video or multiple videos with the framework described in Section IV. On the mobile phone the user can choose from a set of preprocessed grapevines to visualize the cut positions and further instructions overlaid on the camera feed as can be seen in Fig. 12. As the user is moving, it is necessary to track the camera position of the mobile phone to ensure that the visualization is always placed at the correct position with respect to the selected grapevine. The six degree of freedom camera pose tracking consists of two parts. In the initialization stage a keyframe with known camera pose is matched to the current video frame. In subsequent video frames, the camera position is updated via frame-to-frame tracking. The application does not only require a convenient keyframe for initialization but also a selected set of 3D points from the point cloud of the grapevine. For each grapevine, the corresponding tracking data and additional files for visualization, e.g. graph model and cut positions, are collected and pushed to the phone.

The application is built for Android and has been developed and tested with the Samsung Galaxy S20 FE 5G, which is equipped with the Qualcomm Snapdragon 865 processor. Using the Snapdragon Neural Processing Engine (SNPE) SDK[4] allows us to access the digital signal processor (DSP) and efficiently run neural networks on the mobile phone. In addition, we use the OpenCV for Android SDK for all image-processing–related tasks on the phone and Unity to access the camera of the mobile device and to implement a User Interface (UI) and add visualizations. Finally, the Android Debug Bridge (ADB) is used to transfer data between computer and mobile phone.

The application is facing several challenges. The compute resources on the mobile phone are limited and, at the same time, augmentations are supposed to be rendered in real-time. Furthermore, the app has to work outdoors under varying weather conditions. Because video processing and tracking are independent and do not run on the same platform, the weather and illumination can significantly differ between initial video recording and the time the user is actually going to the field to cut the grapevine. Hence, we have to consider that days or weeks may pass between video recording and tracking.

In order to face these challenges, we use a node tracking pipeline based on a light-weight node detection network optimized to run on the DSP of the mobile phone. Finally, 3D to 2D node correspondences are used to estimate the camera position in each frame (Section V-A). Given the camera position of the current video frame, the cut position and further relevant information can be correctly visualized (Section V-B). For computational efficiency, we use a reduced image resolution at each processing step during tracking and only retain high resolution data for visualization.

A. GRAPEVINE TRACKING

In this section, we present our solution for grapevine tracking. The main objective is the ability to accurately track a grapevine in a computationally efficient manner on a mobile phone. In particular, we face the problem of 6DoF camera pose tracking which can be stated as follows. Given 3D points $\mathbf{X} \in \mathbb{R}^3$ in the object coordinate system and corresponding 2D image projections $\mathbf{x} \in \mathbb{R}^2$ in the camera coordinate system, we compute the camera pose consisting of rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ and translation vector $\mathbf{t} \in \mathbb{R}^3$ such that it describes the best fit of (1).

1) SYSTEM OVERVIEW

As can be derived from (1), the key component to estimate the camera pose is a set of reliable 3D to 2D correspondences $C = \{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}$. In our tracking algorithm they are

[4]SNPE: https://developer.qualcomm.com/sites/default/files/docs/snpe/index.html

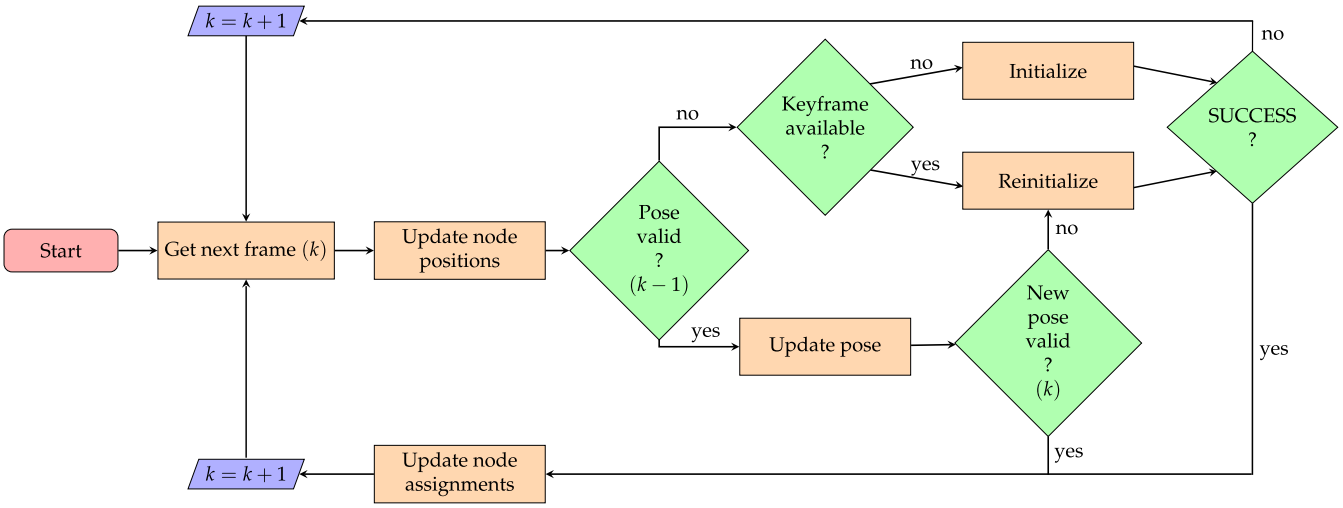


FIGURE 13. Flow diagram of the grapevine tracking algorithm.

based on the grapevine nodes. For one, nodes are the most distinctive points of the grapevine which simplifies matching them between different views. Furthermore, they are well distributed on the grapevine increasing the accuracy of pose estimation and resulting in a precise overlay of virtual content over the whole grapevine structure.

In a preprocessing step, a set of representative 3D node coordinates are extracted from the point cloud. Furthermore, a keyframe with known camera pose is selected for the initialization stage of the tracking. Given a start frame for initialization and a set of 3D node coordinates, our tracking approach follows the process flow outlined in Fig. 13.

As a first step during tracking we apply our node detection network to each new frame arriving from the live feed of the RGB camera. In order to establish relations between nodes of nearby frames, optical flow is used to translate node detections from the previous into the current frame (Section V-A2). Afterwards, the algorithm consists of two parts.

The first part is the initialization stage that is running at the beginning of the tracking based on the start frame and whenever tracking is lost, e.g. due to fast motion. Note that after the first successful initialization and from time to time throughout the tracking, a virtual keyframe is stored. Hence, the virtual keyframe represents the last successfully tracked view and simplifies reinitialization whenever tracking is lost. The initialization stage is further described in Section V-A3.

The second part assumes that tracking has already been successfully initialized. Accordingly, we exploit the information about correspondences between 2D node positions of subsequent frames from optical flow matching to solve a Perspective-n-Point problem and get a pose estimate of the current frame (Section V-A4).

2) NODE DETECTION AND TRACKING

Because classical feature descriptors do not result in robust correspondences in the present of extreme illumination variations, we match correspondences based on grapevine nodes. To predict reliable node positions in any outdoor scenario, we use a neural network for object detection and train it on our dataset that covers a variety of weather conditions. In particular, we use YOLOv5 [85] as base architecture for our use case. YOLOv5 is a single-stage object detection algorithm and known for its fast operating speed that can meet real-time performance requirements while providing a level of accuracy which is comparable to two-stage detection models.

We modify YOLOv5 to focus on small-target detection only and to reduce inference time on the mobile phone. First of all, we choose the nano configuration of YOLOv5. Based on the given YOLOv5n structure, we modify the model in the following ways. In the original architecture the Sigmoid Linear Unit (SiLU) activation function is used after each convolution. However, SiLU is not optimized for the execution with SNPE SDK. Therefore, we replace each SiLU activation function with the more commonly used Leaky Rectified Linear Unit (Leaky ReLU). Although a slight performance drop is expected, exchanging the activation functions will optimize execution of the network on the mobile device. Furthermore, the original YOLOv5 model has three detection heads for small, medium and large objects. We prune the detection heads for medium and large objects to focus on small targets only. The final architecture can be seen in Figure 21 in the Appendix.

Regarding the input data, we chose an image size of resolution 480 $\times$ 480 pixels. This is a trade-off between inference time and the accuracy of node detections. On the one hand, the input image size heavily impacts inference time, on the other hand, dealing with small objects requires a minimum image size to obtain reliable detections.

Due to the previously mentioned network modifications, we train our node detection network from scratch without loading pretrained weights. Once the network is trained, we export and quantize the model for efficient inference on

the mobile phone. In particular, the network is exported to the deep learning container (DLC) format which can be loaded by the SNPE SDK. At runtime, we set an appropriate confidence threshold and use non-maximum suppression (NMS) to get final predictions.

In order to establish node correspondences between subsequent frames, we do not only predict node positions on every frame, but also track the detected nodes using optical flow. Specifically, we use KLT tracking [86] which exploits local optical flow techniques and, thus, significantly improves runtime performance with respect to other optical flow methods, however, restricting matching to small displacements between nearby frames.

3) INITIALIZATION

Initialization is required at the beginning of tracking and whenever tracking is lost. Tracking can be lost, for example, if the user is moving too fast and the displacement between subsequent frames is too large for optical flow matching.

There is always a reference frame necessary to initialize or reinitialize tracking. Specifically, we use template matching between image patches extracted at 2D node positions of the reference frame and node positions of the current live frame. Considering challenging lighting conditions and occlusions, template matching only succeeds if the perspective of the user is close to the one of the reference frame. If a sufficient number of matches are obtained, we can solve a Perspective-n-Point problem within a RANSAC scheme [84] for outlier rejection to estimate the current camera pose.

The initial reference frame is picked from the set of keyframes from the video recording and is selected such that the user can start tracking in a convenient position. After the first successful initialization, the reference frame is directly renewed in order to capture a reference frame that maps current weather conditions. During tracking, the reference frame is constantly renewed at a high rate to always ensure that a perspective close to the last perspective of the user is available for reinitialization.

4) FRAME-TO-FRAME TRACKER

Frame-to-frame tracking is applied as long as the pose from the previous frame is valid. Given our node tracking scheme as described in Section V-A2, we can forward 3D to 2D node correspondences from the previous to the current frame and update the pose accordingly. In order to replenish correspondences at each iteration, we project the complete set of 3D node coordinates into the current frame given the estimated pose and match the projections with current 2D node positions. Finally, we store the resulting assignments for the next iteration. If the pose cannot be estimated because of inaccurate or insufficient matches, we directly try to reinitialize the tracking with the last reference frame.

B. VISUALIZATION

The tracking application is running on a mobile device. Corresponding UI and visualizations can be seen in Fig. 14. The example images are extracted from a screen recording of the mobile phone. In Fig. 14a, the reference frame is shown in the top left corner of the screen in order to simplify initialization. Assuming tracking is initialized and we have a camera pose estimate of the current frame, we can superimpose different augmentations on the grapevine structure. The final visualization covers three categories, either showing graph model and cut positions (14b), fruiting canes and future spurs (14c) or cut positions only (14d).

VI. EVALUATION AND RESULTS

In this section, we provide the evaluation results of our prototype, which includes the assessment of the segmentation network performance as well as the evaluation of pruning suggestions and tracking in both outdoor and indoor settings. These evaluations were conducted by domain experts in the field.

A. EVALUATION OF THE SEMANTIC SEGMENTATION

Image segmentation of dormant grapevines requires handling long and thin branches spanning across large parts of an image as well as small objects (e.g., buds). This, coupled with varying outdoor environments and challenging backgrounds, makes for a difficult computer vision task.

1) QUANTITATIVE EVALUATION

To evaluate our semantic segmentation network, we use mean intersection over union

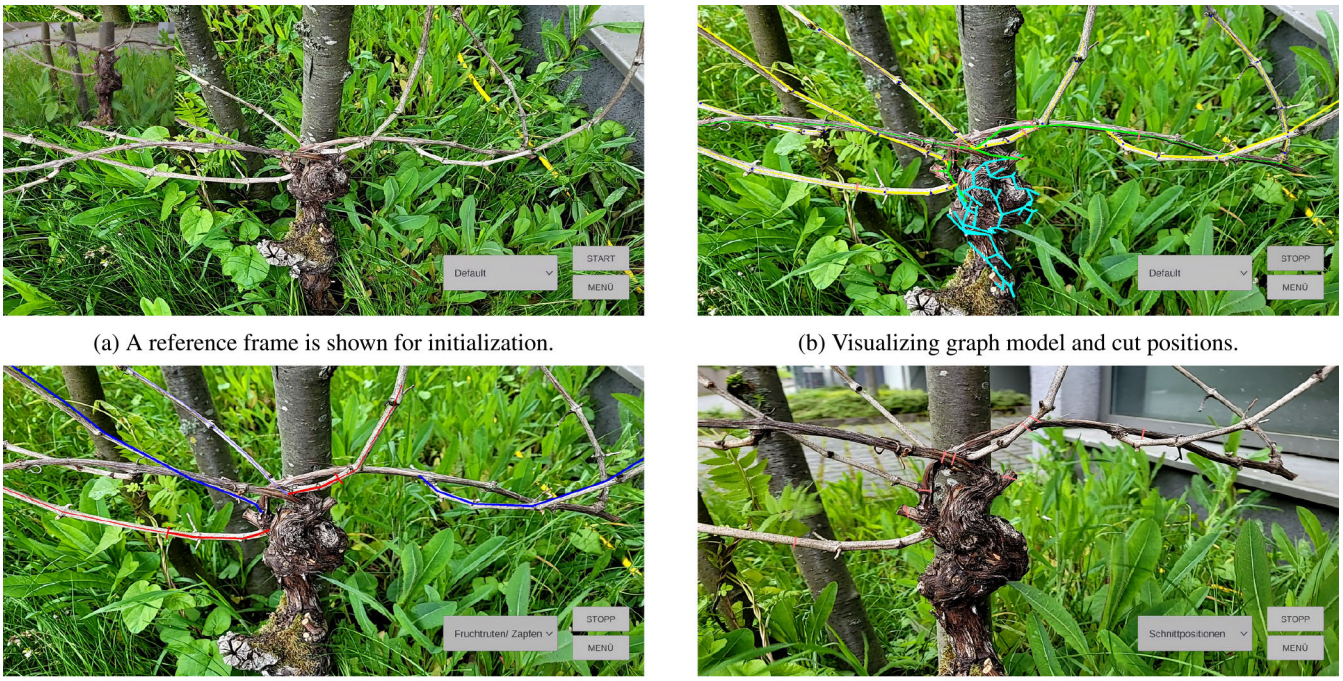
$$mIoU = \frac{1}{C}\sum_{i}^{C} IoU_i = \frac{1}{C}\sum_{i}^{C}\sum_{j}^{S}\frac{TP_{ij}}{TP_{ij}+FP_{ij}+FN_{ij}}, \quad (2)$$

as it is a standard image segmentation metric. It is the mean across all C classes and within each class i across all S samples. TP_{ij}, FP_{ij} and FN_{ij} are the number of true positive, false positive and false negative pixels for class i in sample j. Table 2 shows the results of DDRNet on our test dataset. Classes like stem, one-year-old and two-year-old wood that cover a larger area in the image are detected well. The performance on classes that are rare or have small regions such as those in the secondary mask, is lacking. Small objects like buds can also disappear entirely in the low-resolution high-level layers of the network. As can be seen in Table 3, the

$$\text{precision}_i = \frac{\sum_j^S TP_{ij}}{\sum_j^S TP_{ij} + \sum_j^S FP_{ij}} \quad (3)$$

of the secondary classes is comparable to that of the primary ones. However, they often remain undetected, as indicated by their low recall, defined as

$$\text{recall}_i = \frac{\sum_j^S TP_{ij}}{\sum_j^S TP_{ij} + \sum_j^S FN_{ij}}. \quad (4)$$



(a) A reference frame is shown for initialization.

(b) Visualizing graph model and cut positions.

(c) Visualizing fruiting canes and future spurs.

(d) Only visualizing cut positions.

FIGURE 14. Different example images showing UI and visualizations of the mobile AR app. After successful initialization, the user can choose between three types of visualization.

TABLE 2. IoU scores per class and mIoU per mask (bold). Classes in the secondary mask achieve far lower scores than those in the primary mask. This is likely due to the large imbalance in area between them.

Class	(m)IoU
one year	0.48
two year	0.45
stem	0.61
nodes	0.23
wire	0.39
primary classes	**0.43**
dry	0.08
cut wounds	0.15
buds	0.03
secondary classes	**0.09**

TABLE 3. Comparison of precision and recall for individual secondary classes and averaged across primary classes (bold).

Class	Precision	Recall
primary classes	**0.49**	**0.76**
dry	0.44	0.09
cut wounds	0.54	0.17
buds	0.45	0.03

2) QUALITATIVE EVALUATION

We have identified some typical failure cases. Fig. 15a shows a sample where a neighboring plant and the wooden trellis pole next to it are segmented. This can lead to issues in the graph generation if branches from neighboring plants reach across the plant that is currently being processed as they may get added to the graph. Further, in the top center of this image there is a branch that was misclassified as a wire (purple). This likely happens because wires, similar to thin one-year-old branches, exhibit few texture features.

In Fig. 15b, we can see an example of two-year-old wood and stem being confused by the network. This is indicated by alternating spots of green and yellow. In line with the results from Table 3, buds, cut wounds and dry segments are rarely found in either sample.

B. OUTDOOR EVALUATION

The prototype was tested and evaluated by winemakers in their vineyards in the past pruning season (February - March 2023). In this section, we present the evaluation results for pruning suggestions and tracking.

We considered the pruning suggestions for a total of 148 grapevines from 9 different vineyards under diverse outdoor conditions. A subset of these grapevines, namely 106 plants, was also evaluated regarding tracking. It is crucial to note that variations in results among different vineyards could arise from the distinct weather and lighting conditions prevalent in various locations. Furthermore, the appearance of grapevines can vary due to factors such as previous pruning techniques, plant age, or grape variety.

1) RESULTS: PRUNING SUGGESTIONS

We asked winemakers to evaluate pruning suggestions based on the correctness of the detected essential branches, such as fruiting canes and future spurs, for each plant. Overall, we assess 4 suggestions per plant for 148 plants. The assessments were graded as

(a) Background plant (including wooden pole) was segmented.

(b) Confusion between two year old wood (green) and stem (yellow) in the center.

FIGURE 15. Examples of common shortcomings of the semantic segmentation. The colors are those shown in Fig. 5.

TABLE 4. The outdoor evaluation results for pruning suggestions. The winemakers were asked to assess the correctness of detected essential branches, such as fruiting canes and future spurs, for each plant, i.e. 4 suggestions per plant. The suggestions were graded as good, neutral, poor, or fatal. Overall, the evaluation includes the grades for 4 × 148 suggestions. Note that a few plants could not be evaluated due to various local conditions, i.e. poor visualization caused by lighting conditions or occlusions.

Grade	Fruiting Cane 1	Fruiting Cane 2	Future Spur 1	Future Spur 2	Overall (%)
good	70	38	88	74	45.61
neutral	18	36	20	20	15.88
poor	28	20	20	29	16.39
fatal	30	45	20	24	20.10
not evaluated	2	9	-	1	2.03

- *good*: a pruning suggestion is correct;
- *neutral*: a pruning suggestion is correct, but not optimal, i.e. a better branch exists;
- *poor*: a wrong suggestion, but not harmful for the grapevine;
- *fatal*: a wrong suggestion, which has the potential to hinder the grapevine growth.

Table 4 shows the results of the expert evaluation. Slightly less than half of the pruning suggestions were graded as good, indicating correct cutting suggestions for future spurs or accurate identification of fruiting canes which should not be cut this year. 15.88% of the suggestions received a grade of neutral, representing non-optimal but still acceptable choices. This was often observed in the case of complex plants where multiple decisions are possible. Additionally, around 16.39% of the suggestions were classified as poor, indicating incorrect choices that may impact grapevine growth, but not in a severe way.

However, 20.1% of all evaluated suggestions were considered as fatal, signifying solutions that are harmful to the plant growth and could negatively impact future yields. The presence of fatal suggestions can be attributed to several reasons, some of which have been identified and addressed in the subsequent version of our prototype.

TABLE 5. The evaluation results for tracking. The winemakers were asked to assess precision of the virtual overlay, tracking speed and visibility of augmentations. Each category was graded as good, neutral, poor or fatal. Overall, the evaluation includes the 3 grades for 106 plants. Note that a few grapevines could not be evaluated with respect to speed and visibility.

Grade	Precision	Speed	Visibility	Overall (%)
good	56	23	63	44.65
neutral	27	55	21	32.39
poor	6	1	4	3.46
fatal	17	17	16	15.72
not evaluated	-	10	2	3.77

2) RESULTS: TRACKING

The prototype can only be evaluated during pruning season, therefore, the last outdoor evaluation of the system took place in winter 2023. At that time, the mobile app development was still ongoing. Hence, an intermediate version of the mobile AR app as described in Section V-A was used for this evaluation. Nonetheless, the main aspects of the algorithm such as node detection and tracking had already been implemented. Table 5 summarizes the evaluation results of 106 plants.

Overall, the winemakers were able to run the tracking for 77.04% of the examples. For 19.18% of the plants, the tracking was either rated as bad or did not work and there were 3.77% without a result. Missing results might have been caused by unstable tracking that got lost too fast and users were not able to judge all categories. Furthermore, tracking for some grapevines likely did not work because of an unsuccessful initialization.

The evaluation is split into three categories. Precision, visibility and speed. Precision describes how good the overlay of virtual content fits the grapevine and is important for the user to correctly identify at which position a branch has to be cut. The second category is visibility. Especially in outdoor environments, augmentations may be hard to recognize because of challenging lighting conditions. For the last category, the users evaluated the speed of the application.

Regarding the location of virtual content and its visibility, more than half of the examples were classified as good. Therefore, we conclude that the precision of node positions that further determine the position of virtual content is sufficient for most cases. Tracking speed however, was rated as good in less than 25% of cases. The most time-consuming part of each tracking cycle is network execution. Having significant latency does not only affect usability, but also makes initialization and frame-to-frame tracking more difficult. If incoming frames cannot be processed in real-time, the displacement of nodes between subsequent frames increases and it is more likely that optical flow matching fails. Because of these results, we heavily focused on network optimization as described in Section V-A2 to improve inference time and the overall speed of the final prototype. With regard to visibility, we adjusted the representation of the 3D grapevine model and the choice of colors as can be seen in Fig. 14.

Besides these quantitative results, winemakers provided us with general feedback on the usability of the mobile app. Particularly, they pointed out that the initialization has two drawbacks. Initialization is very difficult for the user if either an inconvenient start position was chosen or if the weather conditions are challenging, e.g. bright sun resulting in reflections or shadow on the grapevine. An example for difficult initialization conditions can be seen in Fig. 16. In the given example, the camera pose could not be determined. Although reference and live frame show a similar view of the grapevine, template matching yielded only two stable node correspondences.

C. LAB EVALUATION

Due to the limited availability of outdoor evaluations, we requested experts to evaluate an updated version of our prototype using recorded videos, following the procedure described in Section VI-B1. Because of the use of recorded videos, we refer to this as a lab evaluation. A total of 79 grapevines, selected randomly and comprising 4 × 79 suggestions, were evaluated for the updated version of our prototype. For this, we enhanced the graph generation and improved the cutting suggestions by considering more difficult scenarios. However, no changes to the reconstruction stage were made. The evaluation results are shown in Table 6.

In general, we observed an improvement in the evaluation results. Despite the advancements, we still encounter challenges when dealing with difficult cases that arise from complex grapevine appearances or unsatisfactory recordings. It is important to note that these cases were not excluded during the random selection of the evaluation samples. Consequently, these challenging scenarios contribute to the occurrence of poor and fatal results.

VII. DISCUSSION

Based on the findings from the initial testing phase of our prototype, a noticeable proportion of the outcomes observed were fatal. We identified several reasons and addressed some of them in the improved version of our prototype.

TABLE 6. The lab evaluation results for pruning suggestions. The correctness of detected essential branches, such as fruiting canes and future spurs, i.e. 4 suggestions per plant, was evaluated. The suggestions were graded as good, neutral, poor, or fatal. Overall, the evaluation includes the grades for 4 × 79 suggestions.

Grade	Fruiting Cane 1	Fruiting Cane 2	Future Spur 1	Future Spur 2	Overall (%)
good	41	36	48	44	53.48
neutral	16	14	11	19	18.99
poor	11	15	8	6	12.66
fatal	11	14	12	10	14.87

A. FIRST OUTDOOR EVALUATION

At this stage the first prototype which included the entire pipeline was evaluated outdoors. During this testing phase, our primary focus was to assess the functionality, robustness and stability of the system under challenging conditions, rather than solely evaluating the final stage, which is the pruning suggestions. The relatively low percentage of suggestions that could not be evaluated (2.03%) indicates the overall stability of the initial prototype.

B. USABILITY-FOCUSED EVALUATION

Testing and evaluation were carried out by winemakers, who are not AI specialists. Therefore, their main focus was on the visualization aspect, the convenience of the user interface in the mobile app and other usability factors. It is possible that some incorrect suggestions may have arisen due to imprecise localization and visualization, especially under challenging lighting conditions.

C. GRAPEVINE VARIETIES

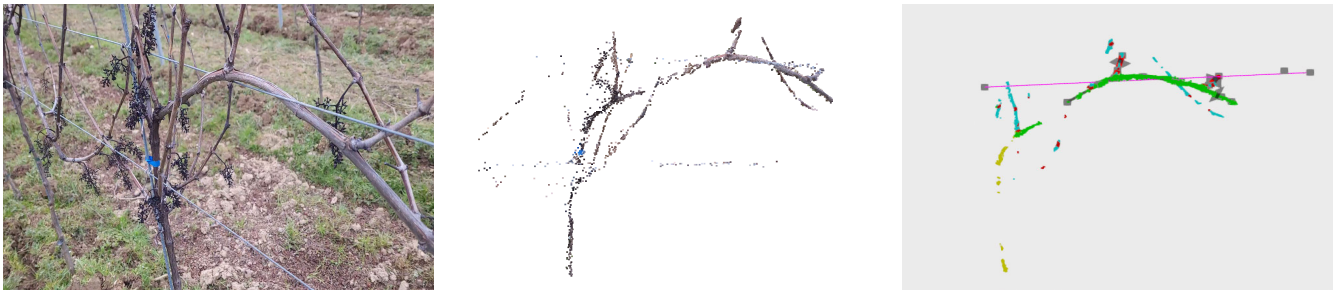
The vineyards, where the prototype was tested, exhibited significant variations in terms of grape variety, grapevine age, previously applied pruning techniques, distance between plants, etc. These factors can also impact the performance of the system. Several vineyards included grapevines that differ considerably from the ones in our dataset. Although we applied proper data augmentation techniques to enhance the robustness of the system, noticeable differences were observed. This variation might have affected the performance of the system, consequently impacting the accuracy of the final results.

An example for results rated as fatal is given in Fig. 17. In this particular case, the grapevine is characterized by a remarkably thin stem and branches, which pose challenges for accurate reconstruction. Consequently, the resulting dense point cloud lacks these crucial structures. As a result, the calculation of cutting suggestions cannot be executed properly. Fig. 17c illustrates the resulting pruning suggestion for a single two-year-old branch. Candidates for fruiting canes and future spurs were not reconstructed and thus remain unidentified.

D. POINT CLOUD ISSUES

The point cloud plays a crucial role in our pipeline, and its quality is essential for obtaining accurate results. Incomplete

FIGURE 16. An example of difficult weather conditions during initialization. The left image shows the start frame for tracking selected from the keyframes of the initial video recording. The right image shows a sample frame from the live camera feed on the day of pruning. The 2D nodes of the start frame are visualized in blue and nodes that are matched between both frames are visualized in red.



(a) A keyframe example. (b) The dense point cloud. (c) The graph and pruning suggestions

FIGURE 17. An example of a fatal result. (a): the keyframe depicts a grapevine with a complex appearance, featuring a thin stem and branches that pose challenges for reconstruction. (b): thin branches are absent from the corresponding dense point cloud. (c): the resulting output shows only partially reconstructed branches. Cutting suggestions for a single two-year-old branch are calculated.

(a) A keyframe example. (b) The dense point cloud. (c) The graph and pruning suggestions.

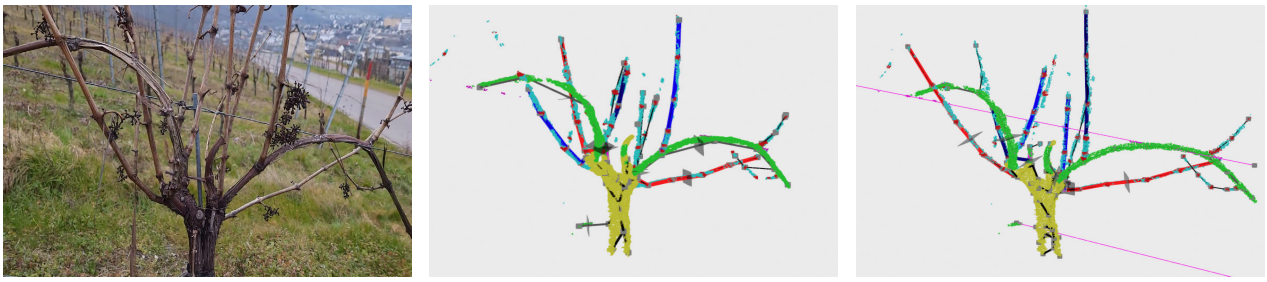
FIGURE 18. An example of an incomplete point cloud caused by a very close view to the plant. (a): an example of the selected keyframes. (b): the reconstructed point cloud. (c): the corresponding graph with pruning suggestions. Here, only the lower part of the grapevine was recorded. As a result, we obtained a point cloud that is unsuitable for the generation of correct pruning suggestions.

and sparse point clouds can have a significant impact on the final output. During the reconstruction stage, various issues can arise that negatively affect the point cloud. For example, inadequate recording with limited perspectives and improper distances can lead to sub-optimal results.

Fig. 18 illustrates a scenario where the grapevine is captured from a very close distance, resulting in only partial registration of the plant. As a result, the generated point cloud does not provide sufficient information about the grapevine structure, making it unsuitable for generating accurate pruning suggestions.

E. TWO-STAGE RANKING

In the initial version of our prototype, we employed a one-stage ranking system for branches. However, we observed that this approach was insufficient in preventing the mixing of candidates for future spurs and fruiting canes. To address this issue, we introduced an additional verification step in the improved version of our prototype. This step helps to avoid such problematic cases. Fig. 19 presents an example where a branch that is suitable for a fruiting cane is mistakenly selected as a future spur that is supposed to be cut. The pruning suggestions from the improved version of



(a) A keyframe example. (b) The initial ptototype. (c) The final prototype.

FIGURE 19. A comparison of the initial and final prototypes. (a): an example of the selected keyframes. (b): the pruning suggestions from the initial prototype, evaluated outdoors, indicate a future spur (red branch) on the left side of the plant, while a fruiting cane is a more suitable option for this branch. (c): the pruning suggestions from the improved version of the prototype correctly select a more suitable branch for the future spur while also identifying a different branch as the fruiting cane.

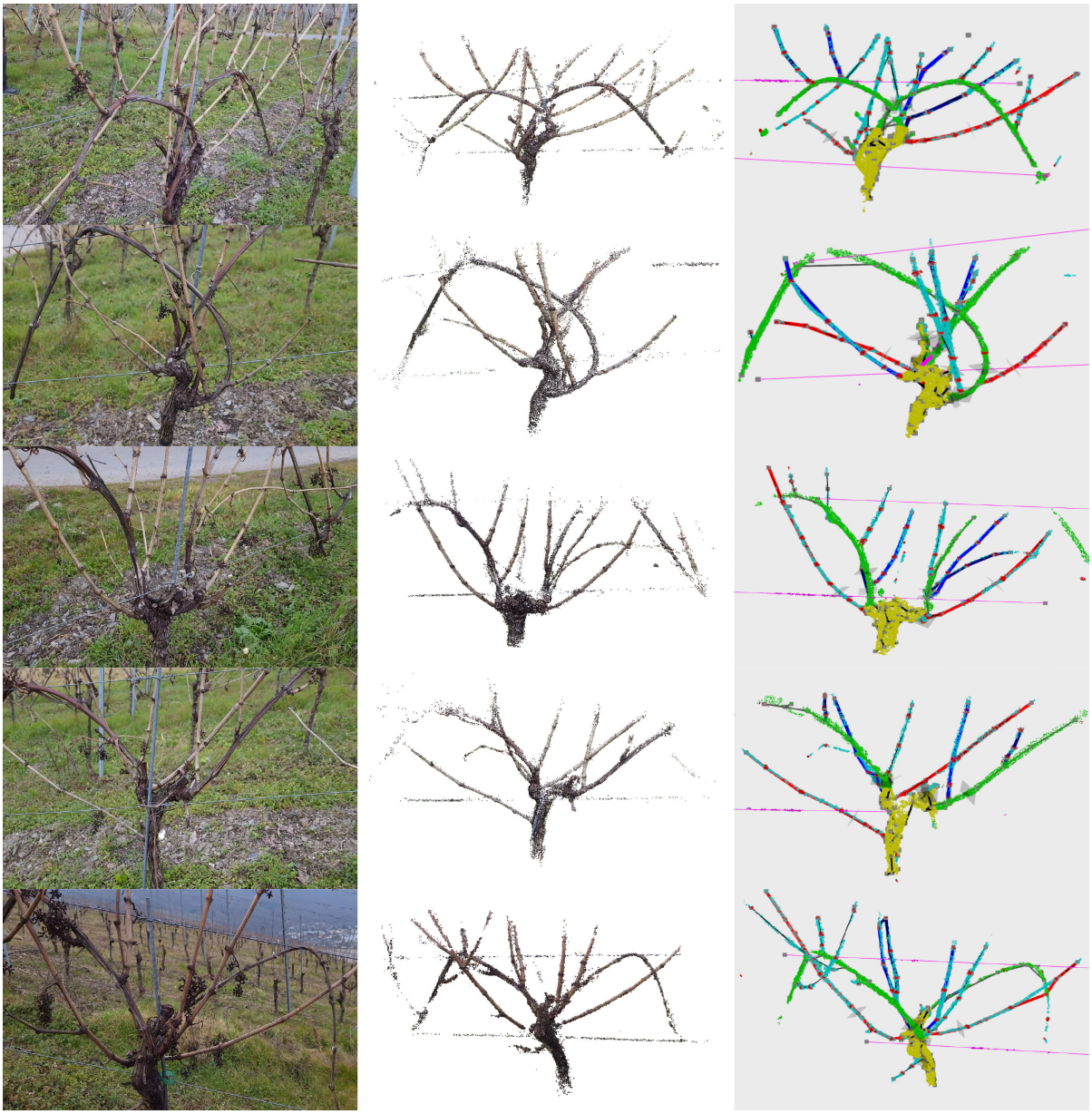


FIGURE 20. Examples with optimal pruning suggestions. **Left:** keyframes. **Middle:** dense point clouds. **Right:** corresponding segmented point clouds with pruning suggestions denoted by gray planes.

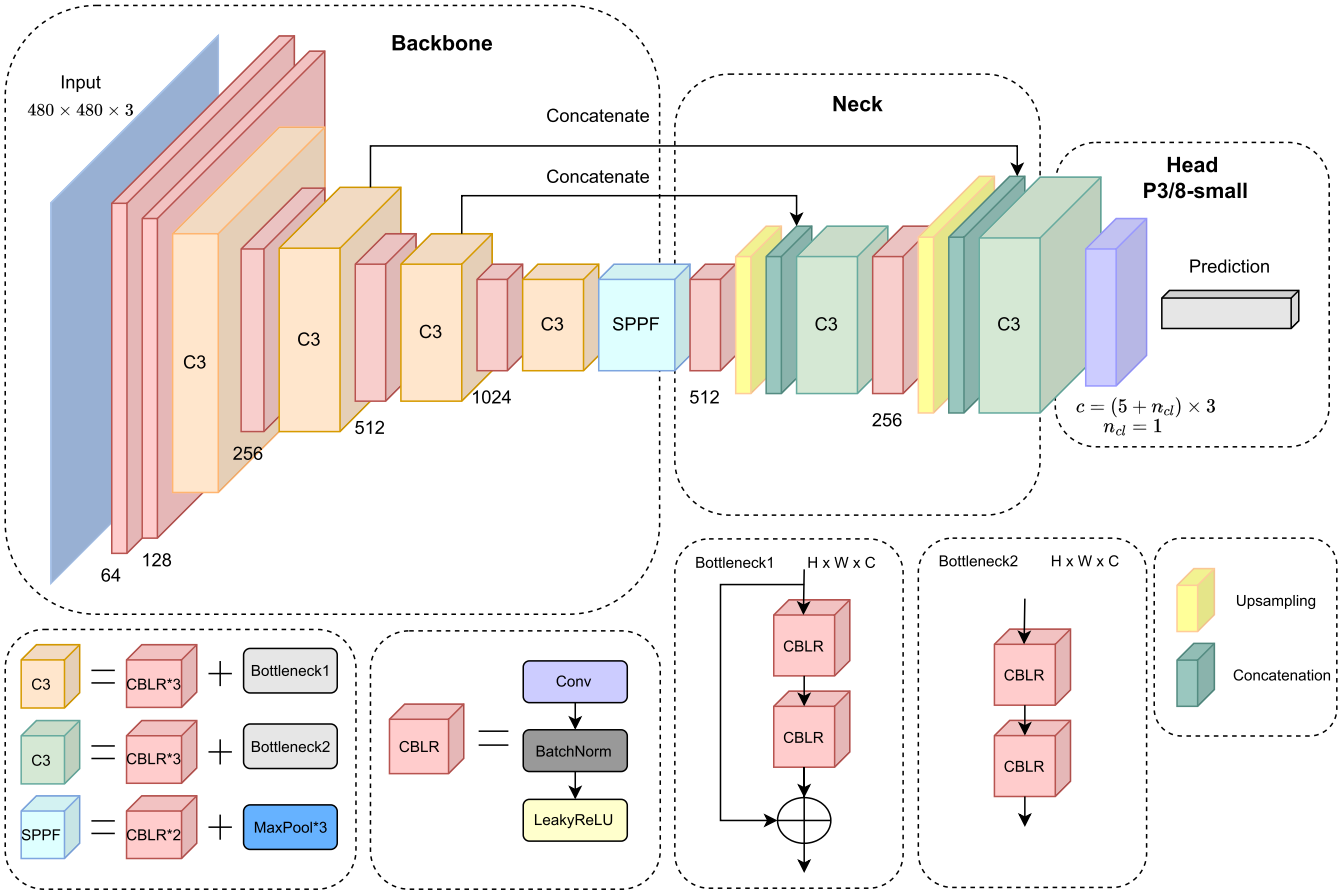


FIGURE 21. The modified architecture of YOLOv5 [85] used in the node detection process during tracking.

the prototype correctly select a more suitable branch for the future spur while also identifying a different branch as the fruiting cane.

VIII. CONCLUSION AND FUTURE WORK

In this work, we presented the collection of a large image segmentation dataset of dormant grapevine plants in the vineyard, a complete pipeline to extract pruning suggestions from a hand-held monocular video of a single plant and a mobile AR application to visualize the cut positions. Our approach uses a dense 3D reconstruction as well as semantic segmentation of multiple images of the same plant. This information is collected in a graph to which we apply a set of pruning rules.

We showed that our system can generate sensible pruning suggestions in over 72% of cases and fails completely in 15% of cases. Further limitations lie in the initialization of the tracking in the AR app as well as the segmentation of small or rare objects like buds.

During testing we identified the 3D reconstruction and the semantic segmentation as the largest contributors to computational cost. Up to this point, our prototypes require an external computer for processing. This limitation creates opportunities for advanced research in the field of high-quality 3D reconstruction as well as semantic segmentation of thin structures, which are optimized for mobile devices. Shifting the focus to a more efficient 3D representation and segmentation network could make a fully mobile prototype feasible.

APPENDIX A EXAMPLES

In the appendix, we present more examples from the final prototype. Fig. 20 shows generated pruning suggestions along with the corresponding keyframes and reconstructed point clouds. The red branches represent potential candidates for future spurs. The blue and dark blue branches indicate the top two candidates for a fruiting cane, i.e. priority 1 and priority 2, respectively. The gray planes mark the designated cut positions.

APPENDIX B DETECTION PROCESS

Figure 21 illustrates the architecture of the modified YOLOv5 [85] used in the node detection process during tracking. We utilized the nano configuration and replaced the Sigmoid Linear Unit (SiLU) activation function by Leaky

Rectified Linear Unit (Leaky ReLU) and used only one detection head focusing on small targets.

DEFINITION OF TERMS

The following terms are used in this work:

basal bud	A bud located at the base of a cane [87].
branch	Or cane, a sequence of connected edges with a unique ending vertex. Each branch is identified by the ID corresponding to the ID of its ending vertex.
bud	A compact node growth that develops into a leaf, or shoot [87].
edge	A segment of a real branch between two vertices.
fruiting cane	Or fruit rod, a one-year-old cane with three or more nodes (in the best case: more than five). It will produce the current season's crop. [87].
future spur	Or future/new cone, a cane pruned to two to four nodes to develop healthy and strong wood that will become fruiting canes after the following winter [87].
node	A node of the grapevine plant, characterized by its unique ID. root the lowest vertex on a stem in the modeled 3D graph.
source	A starting vertex of an edge.
target	An ending vertex of an edge.
vertex	Any vertex of a graph that does not necessarily correspond to a feature on the grapevine plant.

ACKNOWLEDGMENT

The authors would like to thank Carolin Horst for sharing her domain knowledge in viticulture with them.

REFERENCES

[1] M. Simonit, *Simonit&Sirch's Guyot Methodology: The Vine Pruning Manual to Limit Trunk Diseases*. Verona, Italy: L'Informatore Agrario, 2019.

[2] R. Dara, S. M. H. Fard, and J. Kaur, "Recommendations for ethical and responsible use of artificial intelligence in digital agriculture," *Frontiers Artif. Intell.*, vol. 5, Jul. 2022, Art. no. 884192.

[3] P. Arora and A. Jain, "The role of deep learning in agriculture: A review," in *Proc. AIP Conf.*, vol. 2555, Oct. 2022, Paper 050032.

[4] F. Palacios, M. P. Diago, P. Melo-Pinto, and J. Tardaguila, "Early yield prediction in different grapevine varieties using computer vision and machine learning," *Precis. Agricult.*, vol. 24, no. 2, pp. 407–435, Aug. 2022.

[5] F. Palacios, G. Bueno, J. Salido, M. P. Diago, I. Hernández, and J. Tardaguila, "Automated grapevine flower detection and quantification method based on computer vision and deep learning from on-the-go imaging using a mobile sensing platform under field conditions," *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105796.

[6] S. Bargoti and J. Underwood, "Deep fruit detection in orchards," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3626–3633.

[7] C. C. Foong, G. K. Meng, and L. L. Tze, "Convolutional neural network based rotten fruit detection using ResNet50," in *Proc. IEEE 12th Control Syst. Graduate Res. Colloq. (ICSGRC)*, Aug. 2021, pp. 75–80.

[8] N. Ismail and O. A. Malik, "Real-time visual inspection system for grading fruits using computer vision and deep learning techniques," *Inf. Process. Agricult.*, vol. 9, no. 1, pp. 24–37, Mar. 2022.

[9] S. Bobde, S. Jaiswal, P. Kulkarni, O. Patil, P. Khode, and R. Jha, "Fruit quality recognition using deep learning algorithm," in *Proc. Int. Conf. Smart Gener. Comput., Commun. Netw.*, Oct. 2021, pp. 1–5.

[10] J. Jun, J. Kim, J. Seol, J. Kim, and H. I. Son, "Towards an efficient tomato harvesting robot: 3D perception, manipulation, and end-effector," *IEEE Access*, vol. 9, pp. 17631–17640, 2021.

[11] H. Kinjo, N. Oshiro, and S. C. Duong, "Fruit maturity detection using neural network and an odor sensor: Toward a quick detection," in *Proc. 10th Asian Control Conf. (ASCC)*, May 2015, pp. 1–4.

[12] M. A. Matboli and A. Atia, "Fruit disease's identification and classification using deep learning model," in *Proc. 2nd Int. Mobile, Intell., Ubiquitous Comput. Conf. (MIUCC)*, May 2022, pp. 432–437.

[13] N. Saranya, L. Pavithra, N. Kanthimathi, B. Ragavi, and P. Sandhiyadevi, "Detection of banana leaf and fruit diseases using neural networks," in *Proc. 2nd Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Jul. 2020, pp. 493–499.

[14] T. Botterill, S. Paulin, R. Green, S. Williams, J. Lin, V. Saxton, S. Mills, X. Chen, and S. Corbett-Davies, "A robot system for pruning grape vines," *J. Field Robot.*, vol. 34, no. 6, pp. 1100–1122, Sep. 2017.

[15] J. Fourie, C. Bateman, J. Hsiao, K. Pahalawatta, O. Batchelor, P. E. Misse, and A. Werner, "Towards automated grape vine pruning: Learning by example using recurrent graph neural networks," *Int. J. Intell. Syst.*, vol. 36, no. 2, pp. 715–735, Feb. 2021.

[16] J. Zhang, L. He, M. Karkee, Q. Zhang, X. Zhang, and Z. Gao, "Branch detection for apple trees trained in fruiting wall architecture using depth features and regions-convolutional neural network (R-CNN)," *Comput. Electron. Agricult.*, vol. 155, pp. 386–393, Dec. 2018.

[17] S. Amatya, M. Karkee, A. Gongal, Q. Zhang, and M. D. Whiting, "Detection of cherry tree branches with full foliage in planar architecture for automated sweet-cherry harvesting," *Biosyst. Eng.*, vol. 146, pp. 3–15, Jun. 2016.

[18] T. Gentilhomme, M. Villamizar, J. Corre, and J.-M. Odobez, "Towards smart pruning: ViNet, a deep-learning approach for grapevine structure estimation," *Comput. Electron. Agricult.*, vol. 207, Apr. 2023, Art. no. 107736.

[19] M. Fernandes, A. Scaldaferri, G. Fiameni, T. Teng, M. Gatti, S. Poni, C. Semini, D. Caldwell, and F. Chen, "Grapevine winter pruning automation: On potential pruning points detection through 2D plant modeling using grapevine segmentation," in *Proc. IEEE 11th Annu. Int. Conf. Cyber Technol. Autom., Control, Intell. Syst. (CYBER)*, Jul. 2021, pp. 13–18.

[20] S. Katyara, F. Ficuciello, D. G. Caldwell, F. Chen, and B. Siciliano, "Reproducible pruning system on dynamic natural plants for field agricultural robots," in *Human-Friendly Robotics 2020*. Cham, Switzerland: Springer, 2021, pp. 1–15.

[21] H. Medeiros, D. Kim, J. Sun, H. Seshadri, S. A. Akbar, N. M. Elfiky, and J. Park, "Modeling dormant fruit trees for agricultural automation," *J. Field Robot.*, vol. 34, no. 7, pp. 1203–1224, Oct. 2017.

[22] A. You, N. Parayil, J. G. Krishna, U. Bhattarai, R. Sapkota, D. Ahmed, M. Whiting, M. Karkee, C. M. Grimm, and J. R. Davidson, "An autonomous robot for pruning modern, planar fruit trees," 2022, *arXiv:2206.07201*.

[23] A. Tabb and H. Medeiros, "A robotic vision system to measure tree traits," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 6005–6012.

[24] Y. Majeed, J. Zhang, X. Zhang, L. Fu, M. Karkee, Q. Zhang, and M. D. Whiting, "Apple tree trunk and branch segmentation for automatic trellis training using convolutional neural network based semantic segmentation," *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 75–80, 2018.

[25] Z. Zheng, J. Xiong, H. Lin, Y. Han, B. Sun, Z. Xie, Z. Yang, and C. Wang, "A method of green citrus detection in natural environments using a deep convolutional neural network," *Frontiers Plant Sci.*, vol. 12, Sep. 2021, Art. no. 705737.

[26] M. Fernandes, A. Scaldaferri, P. Guadagna, G. Fiameni, T. Teng, M. Gatti, S. Poni, C. Semini, D. Caldwell, and F. Chen, "Towards precise pruning points detection using semantic-instance-aware plant models for grapevine winter pruning automation," 2021, *arXiv:2109.07247*.

[27] Q. Wang and Q. Zhang, "Three-dimensional reconstruction of a dormant tree using RGB-D cameras," in *Proc. Amer. Soc. Agricult. Biol. Eng. Annu. Int. Meeting*, vol. 2, Jan. 2013, p. 1.

[28] M. Karkee and B. Adhikari, "A method for three-dimensional reconstruction of apple trees for automated pruning," *Trans. ASABE*, vol. 58, pp. 565–574, Jun. 2015.

[29] A. You, C. Grimm, A. Silwal, and J. R. Davidson, "Semantics-guided skeletonization of upright fruiting offshoot trees for robotic pruning," *Comput. Electron. Agricult.*, vol. 192, Jan. 2022, Art. no. 106622.

[30] N. M. Elfiky, S. A. Akbar, J. Sun, J. Park, and A. Kak, "Automation of dormant pruning in specialty crop production: An adaptive framework for automatic reconstruction and modeling of apple trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 65–73.

[31] Y. Fu, Y. Xia, H. Zhang, M. Fu, Y. Wang, W. Fu, and C. Shen, "Skeleton extraction and pruning point identification of jujube tree for dormant pruning using space colonization algorithm," *Frontiers Plant Sci.*, vol. 13, Jan. 2023, Art. no. 1103794.

[32] W. Ji, X. Meng, Z. Qian, B. Xu, and D. Zhao, "Branch localization method based on the skeleton feature extraction and stereo matching for apple harvesting robot," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 3, May 2017, Art. no. 172988141770527.

[33] C. A. Díaz, D. S. Pérez, H. Miatello, and F. Bromberg, "Grapevine buds detection and localization in 3D space based on structure from motion and 2D image classification," *Comput. Ind.*, vol. 99, pp. 303–312, Aug. 2018.

[34] W. V. Marset, D. S. Pérez, C. A. Díaz, and F. Bromberg, "Towards practical 2D grapevine bud detection with fully convolutional networks," *Comput. Electron. Agricult.*, vol. 182, Mar. 2021, Art. no. 105947.

[35] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Dec. 2009.

[36] F. Okura, "3D modeling and reconstruction of plants and trees: A cross-cutting review across computer graphics, vision, and plant phenotyping," *Breeding Sci.*, vol. 72, no. 1, pp. 31–47, 2022.

[37] N. Kochi, S. Isobe, A. Hayashi, K. Kodama, and T. Tanabata, "Introduction of all-around 3D modeling methods for investigation of plants," *Int. J. Autom. Technol.*, vol. 15, no. 3, pp. 301–312, May 2021.

[38] S. Paulus, "Measuring crops in 3D: Using geometry for plant phenotyping," *Plant Methods*, vol. 15, no. 1, pp. 1–13, Sep. 2019, Art. no. 103.

[39] X. Wang, D. Singh, S. Marla, G. Morris, and J. Poland, "Field-based high-throughput phenotyping of plant height in sorghum using different sensing technologies," *Plant Methods*, vol. 14, no. 1, pp. 1–16, Jul. 2018, Art. no. 53.

[40] M. Disney, "Terrestrial LiDAR: A three-dimensional revolution in how we look at trees," *New Phytologist*, vol. 222, no. 4, pp. 1736–1741, Jun. 2019.

[41] Y. Wang, W. Wen, S. Wu, C. Wang, Z. Yu, X. Guo, and C. Zhao, "Maize plant phenotyping: Comparing 3D laser scanning, multi-view stereo reconstruction, and 3D digitizing estimates," *Remote Sens.*, vol. 11, no. 1, p. 63, Dec. 2018.

[42] S. Soatto and R. Brockett, "Optimal structure from motion: Local ambiguities and global estimates," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jul. 1998, pp. 282–288.

[43] S. N. Sinha, *Multiview Stereo*. Boston, MA, USA: Springer, 2014, pp. 516–522.

[44] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 501–518.

[45] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4104–4113.

[46] C. Griwodz, S. Gasparini, L. Calvet, P. Gurdjos, F. Castan, B. Maujean, G. De Lillo, and Y. Lanthony, "AliceVision meshroom: An open-source 3D reconstruction pipeline," in *Proc. 12th ACM Multimedia Syst. Conf.*, Jun. 2021, pp. 241–247.

[47] H. Scharr, M. Minervini, A. P. French, C. Klukas, D. M. Kramer, X. Liu, I. Luengo, J.-M. Pape, G. Polder, D. Vukadinovic, X. Yin, and S. A. Tsaftaris, "Leaf segmentation in plant phenotyping: A collation study," *Mach. Vis. Appl.*, vol. 27, no. 4, pp. 585–606, May 2016.

[48] M. Alessandrini, R. C. F. Rivera, L. Falaschetti, D. Pau, V. Tomaselli, and C. Turchetti, "A grapevine leaves dataset for early detection and classification of ESCA disease in vineyards through machine learning," *Data Brief*, vol. 35, Apr. 2021, Art. no. 106809.

[49] L. Rossi, M. Valenti, S. E. Legler, and A. Prati, "LDD: A grape diseases dataset detection and instance segmentation," in *Image Analysis and Processing—ICIAP 2022*, Cham, Switzerland: Springer, 2022, pp. 383–393.

[50] L. Zabawa, A. Kicherer, L. Klingbeil, R. Töpfer, H. Kuhlmann, and R. Roscher, "Counting of grapevine berries in images via semantic segmentation using convolutional neural networks," *ISPRS J. Photogramm. Remote Sens.*, vol. 164, pp. 73–83, Jun. 2020.

[51] R. Rudolph, K. Herzog, R. Töpfer, and V. Steinhage, "Efficient identification, localization and quantification of grapevine inflorescences and flowers in unprepared field images using fully convolutional networks," *Vitis J. Grapevine Res.*, vol. 58, no. 3, pp. 95–104, Aug. 2019.

[52] T. T. Santos, L. L. de Souza, A. A. dos Santos, and S. Avila, "Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association," *Comput. Electron. Agricult.*, vol. 170, Mar. 2020, Art. no. 105247.

[53] K. D. Apostolidis, T. Kalampokas, T. P. Pachidis, and V. G. Kaburlasos, "Grapevine plant image dataset for pruning," *Data*, vol. 7, no. 8, p. 110, Aug. 2022.

[54] M. Fernandes, P. Guadagna, and A. Santamaria, "Grapevine dataset for plant organ segmentation," Zenodo, Sep. 2021, doi: 10.5281/zenodo.5501784.

[55] Y. Majeed, M. Karkee, Q. Zhang, L. Fu, and M. D. Whiting, "Determining grapevine cordon shape for automated green shoot thinning using semantic segmentation-based deep learning networks," *Comput. Electron. Agricult.*, vol. 171, Apr. 2020, Art. no. 105308.

[56] Y. Majeed, J. Zhang, X. Zhang, L. Fu, M. Karkee, Q. Zhang, and M. D. Whiting, "Deep learning based segmentation for automated training of apple trees on trellis wires," *Comput. Electron. Agricult.*, vol. 170, Mar. 2020, Art. no. 105277.

[57] D. Borrenpohl and M. Karkee, "Automated pruning decisions in dormant sweet cherry canopies using instance segmentation," *Comput. Electron. Agricult.*, vol. 207, Apr. 2023, Art. no. 107716.

[58] C. Bento, P. R. da Cunha, and J. Barata, "Cultivating sociomaterial transformations in agriculture 4.0: The case of precision viticulture," in *Proc. 25th Americas Conf. Inf. Syst. (AMCIS)*. Atlanta, GA, USA: Association for Information Systems (AIS), Aug. 2019.

[59] J. Huuskonen and T. Oksanen, "Soil sampling with drones and augmented reality in precision agriculture," *Comput. Electron. Agricult.*, vol. 154, pp. 25–35, Nov. 2018.

[60] Z. Zhao, W. Yang, W. Chinthammit, R. Rawnsley, P. Neumeyer, and S. Cahoon, "A new approach to utilize augmented reality on precision livestock farming," in *Proc. Int. Conf. Artif. Reality Telexistence Eurograph. Symp. Virtual Environ. (ICAT-EGVE)*, pp. 185–188, 2017.

[61] M. Caria, G. Sara, G. Todde, M. Polese, and A. Pazzona, "Exploring smart glasses for augmented reality: A valuable and integrative tool in precision livestock farming," *Animals*, vol. 9, no. 11, p. 903, Nov. 2019.

[62] W. Hurst, F. R. Mendoza, and B. Tekinerdogan, "Augmented reality in precision farming: Concepts and applications," *Smart Cities*, vol. 4, no. 4, pp. 1454–1468, Dec. 2021.

[63] X. Yang, L. Zhou, H. Jiang, Z. Tang, Y. Wang, H. Bao, and G. Zhang, "Mobile3DRecon: Real-time monocular 3D reconstruction on a mobile phone," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 12, pp. 3446–3456, Dec. 2020.

[64] J. Valentin et al., "Depth from motion for smartphone AR," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–19, Dec. 2018.

[65] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[66] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[67] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2014, pp. 834–849.

[68] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut': Interactive foreground extraction using iterated graph cuts," in *Proc. ACM SIGGRAPH*, New York, NY, USA, 2004, pp. 309–314.

[69] V. Torre and T. A. Poggio, "On edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 2, pp. 147–163, Mar. 1986.

[70] R. Bansal, G. Raj, and T. Choudhury, "Blur image detection using Laplacian operator and open-CV," in *Proc. Int. Conf. Syst. Model. Advancement Res. Trends (SMART)*, Nov. 2016, pp. 63–67.

[71] H. Pan, Y. Hong, W. Sun, and Y. Jia, "Deep dual-resolution networks for real-time and accurate semantic segmentation of traffic scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3448–3460, Mar. 2023.

[72] M. Weber, M. Fürst, and J. M. Zöllner, "Automated focal loss for image based object detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1423–1429.
[73] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.
[74] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, Feb. 2020.
[75] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. J. Argus, and T. Brox, "FreiHAND: A dataset for markerless capture of hand pose and shape from single RGB images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 813–822.
[76] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 413–420.
[77] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[78] J. Dong and S. Soatto, "Domain-size pooling in local descriptors: DSP-SIFT," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Los Alamitos, CA, USA, Jun. 2015, pp. 5097–5106.
[79] P. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *Proc. Brit. Mach. Vis. Conf.*, 2013, p. 13.
[80] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jul. 2005, pp. 807–814.
[81] M. Jancosek and T. Pajdla, "Multi-view reconstruction preserving weakly-supported surfaces," in *Proc. CVPR*, Jun. 2011, pp. 3121–3128.
[82] M. Jancosek and T. Pajdla, "Exploiting visibility information in surface reconstruction to preserve weakly supported surfaces," *Int. Scholarly Res. Notices*, vol. 2014, pp. 1–20, Aug. 2014.
[83] S. Katz, A. Tal, and R. Basri, "Direct visibility of point sets," *ACM Trans. Graph.*, vol. 26, no. 3, p. 24, Jul. 2007.
[84] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[85] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[86] J.-Y. Bouguet, "Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm," *Intel Corp.*, vol. 5, nos. 1–10, p. 4, 2001.
[87] T. Goldammer, *The Grape Grower's Handbook: A Guide to Viticulture for Wine Production*. Centreville, VA, USA: Apex, 2015.

SIMON HÄRING received the B.Sc. and M.Sc. degrees in computer science, with a focus on image processing and computer graphics from the University of Koblenz, in 2018 and 2021, respectively. His master's thesis focused on human action segmentation from skeleton data using transformer networks. Since 2021, he has been a Researcher with Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau (RPTU), Germany. His current research interest includes computer vision, with a focus on detection and segmentation of thin structures.

SOPHIE FOLAWIYO received the bachelor's and master's degrees in electrical and computer engineering from the Technical University of Munich. In 2019, she completed her master's thesis project at a startup company in Stockholm, where she explored deep learning models for 3D shape representation. She worked both in industry as a Development and Data Engineer in the field of motion capture and an Academic Researcher with the University of Kaiserslautern-Landau. Her research interests include deep learning, computer vision, and augmented reality.

MARIIA PODGUZOVA received the M.Sc. degree in computer science from the University of Stuttgart, in 2021. She is currently a Researcher with the University of Kaiserslautern-Landau (RPTU) in collaboration with the German Research Center for Artificial Intelligence (DFKI). Her study focus was machine and deep learning and computer vision. She worked on related projects as an intern and a working student, including uncertainty quantification and calibration for object detector applied in autonomous driving. Her master's thesis was dedicated to the problem of "Image reconstruction from human brain activity by variational autoencoder and adversarial learning." Her research interests include deep learning in computer vision, such as object detection, semantic segmentation, reconstruction, and generative models.

STEPHAN KRAUß received the Diploma degree in computer science from the Technical University Dresden, in 2010. He is currently a Researcher with the German Research Center for Artificial Intelligence (DFKI), Kaiserslautern. His research interests include computer vision and deep learning, with a focus on efficient neural network architectures for the use in mobile and edge devices.

DIDIER STRICKER led the Department of Virtual and Augmented Reality, Fraunhofer Institute for Computer Graphics, Darmstadt, Germany, from 2002 to 2008. He is currently a Professor with the Department of Computer Science, Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau (RPTU), Germany. He is also a Scientific Director of the German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, where he leads the Augmented Vision Research Group. His research interests include 3D computer vision, autonomous driving, wearable health, augmented reality applications, and deep learning. He received the Innovation Prize from the German Society of Computer Science, in 2006. He serves as a reviewer for noteworthy journals in the area of VR/AR and computer vision.