




Semi-supervised deep learning and low-cost cameras for the semantic segmentation of natural images in viticulture

A. Casado-García¹ · J. Heras¹ · A. Milella² · R. Marani² 

Accepted: 28 May 2022 / Published online: 21 June 2022
© The Author(s) 2022

Abstract

Automatic yield monitoring and in-field robotic harvesting by low-cost cameras require object detection and segmentation solutions to tackle the poor quality of natural images and the lack of exactly-labeled datasets of consistent sizes. This work proposed the application of deep learning for semantic segmentation of natural images acquired by a low-cost RGB-D camera in a commercial vineyard. Several deep architectures were trained and compared on 85 labeled images. Three semi-supervised learning methods (PseudoLabeling, Distillation and Model Distillation) were proposed to take advantage of 320 non-annotated images. In these experiments, the DeepLabV3+ architecture with a ResNext50 backbone, trained with the set of labeled images, achieved the best overall accuracy of 84.78%. In contrast, the Manet architecture combined with the EfficientnetB3 backbone reached the highest accuracy for the bunch class (85.69%). The application of semi-supervised learning methods boosted the segmentation accuracy between 5.62 and 6.01%, on average. Further discussions are presented to show the effects of a fine-grained manual image annotation on the accuracy of the proposed methods and to compare time requirements.

Keywords Semantic segmentation · Semi-supervised learning · Grape bunches · Natural images · Agricultural robot sensing

Introduction

Sustainability is a crucial goal that involves ecological, economic and social concerns to impact the health of present and future societies. Scientific progress has developed new automatic tools to assist the human workforce by integrating artificial intelligence and robotics to meet such high-level needs. These efforts affect all production fields but, significantly, agriculture, whose improvement needs to face sustainability-related topics, such as

✉ R. Marani
roberto.marani@stiima.cnr.it

¹ Department of Mathematics and Computer Science, University of La Rioja, Logroño, Spain

² Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council of Italy, Bari, Italy

finite resource management, yield optimization and pest control. In general, every sustainable goal can require actual crop monitoring by implementing low-cost technologies (cameras) and reliable methodologies (machine and deep learning techniques) in engineered solutions (Saleem et al., 2021). These requirements translate into the need for developing image acquisition and processing systems for extracting helpful information for the farmer. At a low level, systems must identify specific targets by applying semantic inference mechanisms, including image classification or segmentation.

In general, crop monitoring without physical contact of the targets can be clustered in remote and proximal sensing, depending on the sensor-plant distance and, thus, the level of details of the achievable information. Remote sensing typically refers to aerial imaging from satellites, unmanned aerial vehicles (UAVs) or airplanes. UAVs are equipped with imaging sensors, such as hyperspectral, LIDAR and RGB cameras (Adão et al., 2017; Kim et al., 2019), to compute vegetation indicators, e.g. the normalized difference vegetation index (NDVI) or canopy size and volume (Zhou et al., 2020) or to create semantic maps of the fields (Dyson et al., 2019; Guo et al., 2018; Osco et al., 2021; Wu et al., 2019; Yang et al., 2020a). In proximal sensing, acquisitions are taken from the ground, close to the target, and with more details. Typical sensors include color, hyperspectral and infrared (IR) thermal cameras and LIDAR (Das et al., 2015; Tian et al., 2020a), targeted to object segmentation, fruit counting, phenotype analysis, plant classification and disease monitoring (Jiang et al., 2019; Ma et al., 2017; Yang et al., 2020b). In proximal sensing, data can be collected in structured and well-controlled environmental contexts, such as greenhouses (Afonso et al., 2020; Sa et al., 2016), or under excellent acquisition conditions, typically manual, with high-resolution sensors (Mack et al., 2017). Referring to extensive crops, the practical implementation of proximal sensing is achievable through agricultural robots working in-field. However, any approach to extensive monitoring must face the actual problems of in-field raw image data (natural images), such as low resolution, motion blurring, occlusions and uncontrolled lighting conditions.

The processing of natural images captured from ground robotic platforms, and more specifically the semantic segmentation of images, has been proposed mainly for weed detection (Bosilj et al., 2020; Knoll et al., 2018; Milioto et al., 2018; Wang et al., 2020a), even sharing the same input dataset (Chebroly et al., 2017) and a common processing background, centered on deep learning (LeCun et al., 2015). More specifically, input images, which are often reported in terms of NDVI, are processed by convolutional neural networks (CNNs) for pixel or area classification, trained from scratch or by applying transfer learning (Tan et al., 2018). Deep learning is often used to segment objects of interest, such as fruits, leaves, infrastructures (wires and poles) and single branches (Naranjo-Torres et al., 2020; Wosner et al., 2021). In horticulture, several methodologies have been presented for monitoring fruit orchards through flower classification for thinning (Tian et al., 2020b), fruit classification for automatic harvesting (Gao et al., 2020) and segmentation of supporting infrastructures, such as wires (Song et al., 2021).

Automatic procedures for object segmentation are even more attractive in those areas of horticulture of high added value, such as viticulture (Barriguiha et al., 2021). Here, monitoring at the plant scale allows vine-growers to understand possible spatial variabilities and find fine-tuned solutions. For instance, Majeed et al. (2020) presented a ResNet deep residual network and region-based convolutional neural network to detect green shoots in grapevine canopies and precisely segment the trajectories of cordons for thinning purposes. Grape cluster and canopy segmentation using an artificial neural network and a genetic algorithm on images of a publicly available dataset (Berenstein et al., 2010) were proposed by Behroozi-Khazaei and Maleki (2017), while Santos et al. (2020) presented a

comparison of three neural networks for instance segmentation of grape clusters tested on their public dataset (Embrapa Wine Grape Instance Segmentation Dataset - WGISD) of RGB images captured from a mobile robot.

In any of the cases above, all sensors are standard RGB cameras, which provide a flat 2D representation of the targets. In contrast, RGB-D cameras, able to produce three-dimensional (3D) colored models of the crops, can give more information, helpful for fruit monitoring and counting (Fu et al., 2020a). Several technologies, including complex setups of dedicated 3D cameras (Barnea et al., 2016; Gongal et al., 2016) or integrated low-cost consumer-grade cameras, such as the Microsoft Kinect v1 and v2 cameras (Redmond, WA, USA) (Fu et al., 2020b; Nguyen et al., 2016; Paulus et al., 2014; Tao & Zhou, 2017; Zhang et al., 2018), have been used for plant phenotyping, fruit counting and automatic robotic harvesting. Even low-cost stereo cameras, such as those of the Intel Realsense family (R200 and D4xx, Santa Clara, CA, USA), have gained attention in fruit detection and plant phenotyping (Milella et al., 2019) since they can effectively model the outdoors without suffering from illumination variability due to sunlight (Kuan et al., 2019). Several works processing color images acquired by the Intel RealSense R200 and D435 for object segmentation have been presented (Kang & Chen, 2020; Marani et al., 2021; Wang et al., 2020b). Although RGB-D cameras help yield monitoring, output color images are often of low quality and resolution due to the actual scope of such low-cost cameras, which are mainly designed for robot navigation, mapping and manipulation. Natural image segmentation from color data is still an open problem since its effective solution enables the effective use of the depth channel.

In this scenario, this paper extends previous work by Heras et al. (2021) for the exact segmentation of plant leaves and wooden structures (trunks, branches, canes, etc.), artificial infrastructures (poles, ropes, cables, etc.) and fruits. Here, multiple network architectures were compared to find the best solution for natural image segmentation. Even a refined ground truth was considered to further improve the quality of segmentation. The original contribution of the paper is manifold:

1. The analysis of three semi-supervised learning models to contrast the small size of the annotated dataset by taking advantage of unlabeled images;
2. A detailed comparison of several pre-trained deep neural networks (architectures and backbones) for processing images of low-quality, affected by blurring and compression artifacts, as captured by consumer-grade devices mounted onto moving agricultural vehicles;
3. A statistical analysis to identify significant differences among the deep learning models studied and the semi-supervised learning methods;
4. A comprehensive discussion on the quality of manual image annotation and how it can affect segmentation.

Materials and methods

Input dataset

This work tackled the problem of segmentation of single natural images captured in-field by the low-cost consumer-grade Intel Realsense R200 camera (Santa Clara, CA, USA). Semantic segmentation is the classification of every pixel of an image among target classes

of interest. In viticulture, segmenting specific targets, such as leaves, fruits, wooden structures (trunks, branches, canes, etc.) and artificial infrastructures (poles, ropes, cables, etc.) can be the key for yield monitoring and robotic harvesting.

Several techniques for natural image segmentation were tested on the dataset by Marani et al., (2019, 2021). This dataset consists of 405 color images acquired by the Intel Realsense R200 in a vineyard in Switzerland (Räuschling, (N47° 14' 27.6", E8° 48' 25.2")). The camera was mounted on a moving agricultural tractor (Niko Caterpillar, Bühl/Baden, Germany) and acquired lateral views of the line of the grape plants at a distance between 0.8 and 1 m. Under those conditions, every image covered a horizontal field of view between 0.9 and 1.2 m to completely frame every plant in a single image. The tractor moved within lines at an average speed of 1.5 m/s. Image frame rate was then tuned according to the robot speed and the horizontal field of view of the camera to frame the same plant in at least three consecutive captures. A camera frame rate of 5 Hz was enough to produce image overlaps, corresponding to about 0.3 m. The image resolution was limited to 640×480 pixels to match the maximum resolution of the depth data stream. It is worth noticing that, although video sequences were produced to create overlaps, the proposed implementations did not take advantage of object tracking strategies, like the one of Santos et al. (2020). All methodological approaches considered images individually, without managing multiple detections of the same elements. A sample image of the dataset is shown in Fig. 1.

As shown in Fig. 1, the resulting images are poor in detail and clearness. As the effect of the movement of the tractor and the low quality of both camera sensor and optics, images suffer from blurring, soft hue and weak contrast. Moreover, the JPEG compression further decreased the quality of the acquisition. For example, the inset of Fig. 1 shows how similar the appearance of the foreground grape bunches and the background small leaves are.

The automatic segmentation of natural images is achieved by representing them in more descriptive and discriminative feature spaces, learned from actual images, where pixels having similar semantic attributes can be grouped and labeled in different classes.

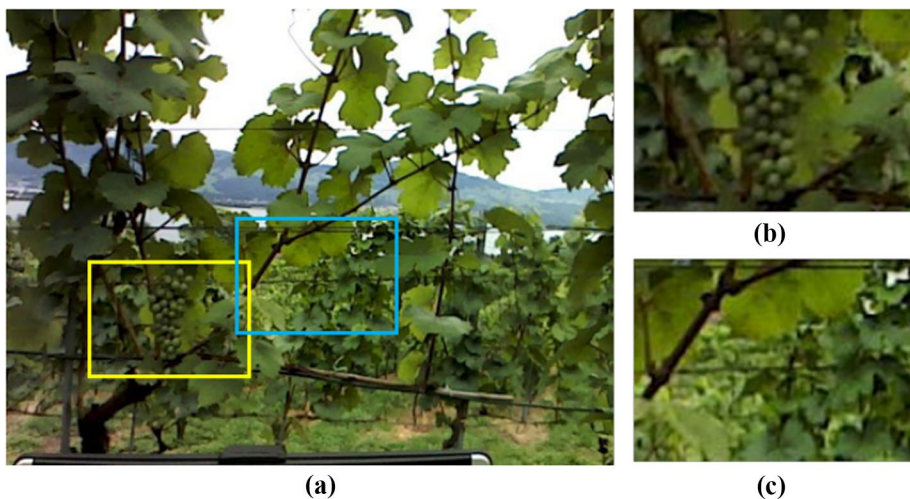


Fig. 1 a A sample color image acquired by the Intel Realsense R200. b and c are magnifications of the area enclosed by the yellow and cyan boxes, respectively

A set of annotated images is thus required to train the model and then evaluate the segmentation results on ground truth.

Manual annotation is a complex, time-demanding and tedious task. For this reason, annotation is typically limited to a small subset of all the images acquired in-field. However, unlabeled images were captured under the same experimental conditions and could give further information to tune the training of the networks using semi-supervised approaches.

The whole dataset of 405 natural color images from the Intel Realsense R200 camera was thus split into two sets of 85 manually annotated images and 320 unlabelled images. The 20–80 proportion was chosen to give more evidence to the improvement of results due to the semi-supervised approaches.

Within these lines, images were processed to segment five classes of interest:

- Bunch: bunches of white grapes;
- Pole: supporting infrastructure made of concrete or metal poles;
- Wood: canes, cordons and trunks of the plant;
- Leaves: canopy leaves of the grape; and,
- Background: the remaining objects framed by the camera, such as the ground, the sky and far grape lines.

Manual annotation was performed twice on the same images to produce two sets of labels:

- Bunch/leaves-detection-oriented (BLDO) labels: BLDO labels were the same as in Milella et al. (2019) and Marani et al., (2019, 2021) and were mainly focused on the bunch and leaves segmentation. The corresponding ground truth was obtained for each image, giving different priority levels to each class. First, bunches were annotated as closed objects, even if their appearance slightly differed from what was expected as the effect of a crossing object or image artifacts. Then, plant leaves, poles and wooden structures were annotated with the same strategy but with decreasing priority levels. The background was the last labeled class, enclosing the remaining pixels; and,
- Object-segmentation-oriented (OSO) labels: OSO labels were created for an object segmentation task, as typically referred to in the corresponding literature. Annotation gave equal priority to every class to label objects as they appeared in the image.

An insight into the difference between the two kinds of labels is shown in Fig. 2. Specifically, all wooden structures or supporting infrastructures, i.e. poles, have more weight and are better detailed since they are no longer included in the leaves class. In the following lines, all analyses were run on BLDO labels for enabling the comparison with previous results in Marani et al., (2019, 2021). Then, further experiments on the best models, trained by OSO labels, are presented to discuss the importance of manual labeling for accurate segmentation results.

The whole datasets, made of in-field natural images and corresponding labels (BLDO and OSO), can be downloaded for further comparative tests at the following webpage: <https://github.com/ispstiima/S3CavVineyardDataset>.

Once the dataset and its annotations have been presented, the following subsections detail the network architectures and the semi-supervised algorithms.

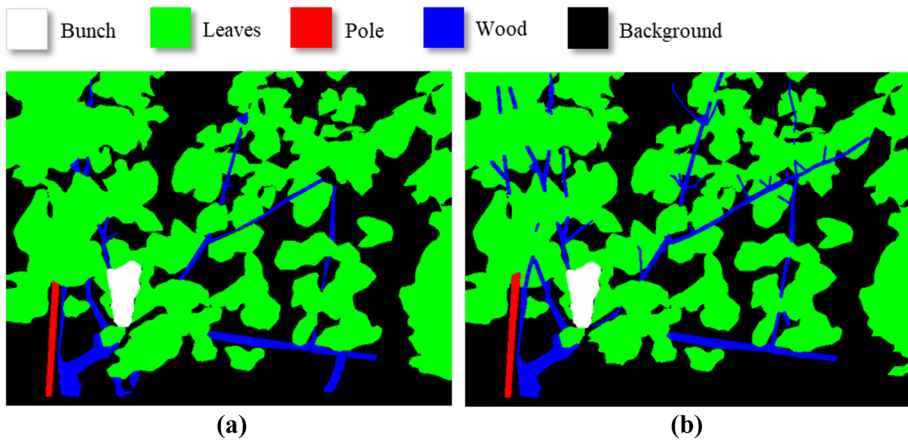


Fig. 2 Annotations of the image in Fig. 1a: **a** bunch/leaves-detection-oriented labels as in Marani et al., (2019, 2021); **b** object-segmentation-oriented labels, resulting in a fine refinement of the BLDO labels in (a)

Semantic segmentation models

As stated in the previous section, the 85 labeled images, split into training and test sets, were used to set up and evaluate the deep segmentation architectures. The training set was used to fine-tune several deep-learning segmentation architectures (Razavian et al., 2014) to produce inference on natural images in the form of output masks. With more details, 13 architectures, summarized in Table 1, were trained. All the selected architectures were based on either fully convolutional networks (FCN) (Long et al., 2015) or encoder-decoder networks (Ronneberger et al., 2015).

FCN architectures extract features from a given image using a backbone of convolutional layers and generate an initial coarse classification map. The classification map is a spatially reduced version of the original image. Then, deconvolutional layers restore the

Table 1 Segmentation architectures and backbones employed in this work

Segmentation architecture	Backbones
Bisenet	Resnet18, Resnet34
CGNet	CGNet
ContextNet	ContextNet
DeepLabV3+	EfficientNet-B3, Resnet50, Resnext50
DenseApp	DenseApp
FPENet	FPENet
HRNet	W30
Manet	EfficientNet-B3, Resnet50, Resnet50
LedNet	Resnet50
PAN	EfficientNet-B3, Resnet50
OCNet	Resnet50
Unet	EfficientNet-B3, Resnet50, Resnet50
Unet++	EfficientNet-B3, Resnet50, Resnet50

original resolution of the classification map to output the final segmentation mask. The main two drawbacks of this architecture are the loss of information when working with high-resolution images and its speed. For tackling the high-resolution problem, in the HRNet architecture (Sun et al., 2019), high-resolution representations were maintained by connecting high-to-low resolution convolutions in parallel and repeatedly conducting multi-scale fusions across parallel convolutions. Atrous convolutions were instead used in the DenseApp architecture (Yang et al., 2018) to face the same resolution issue. For tackling the problem of the high time requirements, the ContextNet architecture (Poudel et al., 2018) used factorized convolution, network compression and pyramid representation, while the CGNet architecture (Wu et al., 2018) employed a context-guided block.

In the encoder-decoder architectures, the encoder is usually made of several convolutional and pooling layers responsible for extracting the features and generating an initial coarse prediction map. In these architectures, the encoder is known as the backbone. The decoder, commonly composed of convolution, deconvolution and/or unpooling layers, is responsible for further processing the initial prediction map, increasing its spatial resolution gradually and generating the final prediction. The Unet architecture (Ronneberger et al., 2015) was the first network to propose an encoder-decoder architecture to perform semantic segmentation in medical contexts. From that seminal work, several variants have been proposed to address the two main limitations of the Unet architecture that are the same as previously mentioned for the FPN architecture: the loss of information when working with high-resolution images and its speed. Regarding the issues related to the use of images of high-resolution:

- the DeepLabV3+ (Chen et al., 2018) architecture introduced the notion of atrous convolutions to extract features at an arbitrary resolution;
- the PAN architecture (Li et al., 2018) adopted global attention upsample module to squeeze high-level context and embedded it into low-level features as guidance;
- the FPENet architecture (Liu & Yin, 2019) defined a MEU module that used attention maps to embed semantic concepts and spatial details to low-level and high-level features; and,
- the Unet++ architecture (Zhou et al., 2018) redesigned the connection between the encoder and the decoder components of the architecture.

Referring to the speed issue:

- the Bisenet architecture (Yu et al., 2018) proposed a fast downsampling strategy to obtain a sufficient receptive field; and,
- the LedNet architecture (Wang et al., 2019) employed an attention pyramid network in the decoder.

All the aforementioned architectures are based on convolutional operations. In addition, two other architectures based on the attention mechanism, namely OCNNet (Yuan & Wang, 2018) and Manet (Li et al., 2020), were considered.

All the architectures with their respective backbones presented in Table 1 were trained using the PyTorch (Paszke et al., 2019) and FastAI (Howard & Gugger, 2020) libraries on an Nvidia RTX 2080 Ti GPU (Santa Clara, CA, USA). The procedure presented in Howard and Gugger (2020) was employed to set the learning rate for the different architectures. Also, early stopping was applied in all the architectures to avoid overfitting.

After training, all the models were then evaluated on the test set of 25 annotated images using the mean segmentation accuracy of the c -th class (MSA_c):

$$MSA_c = \text{mean} \left\{ \frac{TP_c}{n_{obs,c}}, \forall \text{image} \in \text{Dataset} \right\} \quad (1)$$

where TP_c is the number of true positives, i.e. correct pixel labels over the entire population of the c -th class ($n_{obs,c}$) (Marani et al., 2021).

All the code necessary for training the models is available at <https://github.com/ancasag/GrapeBunchSegmentation>.

Semi-supervised learning methods

As stated in the previous section, the dataset contains 320 additional unlabeled images. In this case, semi-supervised learning approaches can help the training phase by adding more information from unlabeled images. For this reason, three semi-supervised learning approaches were employed: PseudoLabeling (Lee, 2013), Distillation (Hinton et al., 2015) and Model distillation (Bucilua et al., 2006). Figure 3 presents a sketch of each of these semi-supervised learning methods.

The pseudoLabeling approach consists of two steps: given a deep learning architecture, a first model is trained using that architecture on a manually labeled dataset to make predictions in an unlabeled dataset; secondly, the manually and automatically-labeled datasets are combined to train a new model using the same previous architecture. This pseudolabeling approach was applied to all the architectures presented in the last section (Table 1).

The distillation approach is similar to pseudolabeling, but in the second step, the trained model might have a different underlying architecture than the model trained on the first step. In this case, all the models of Table 1 were trained using the training procedure presented in the previous sections, but only the best model was used for generating the automatically labeled dataset. Then, both sets (manually and automatically labeled) were combined to re-train all the architectures in Table 1.

Finally, model distillation differs from the distillation approach in producing the automatically labeled dataset. Instead of using a single model for making predictions in an unlabeled dataset, predictions are generated from an ensemble of models. In this approach, the five models with the best total MSA produced the predictions on the unlabeled dataset, which were then combined to create single images. Finally, as in the previous approaches, the manually and automatically-labeled datasets were used to train all the architectures presented in the last section.

Experimental study

In addition to searching for the best-performing model, a statistical study was conducted to determine whether the results obtained with the different semi-supervised learning approaches were statistically significant. To this aim, several null hypothesis tests were performed using the methodology presented by García et al (2010) and Sheskin (2011). In order to choose between a parametric or a non-parametric test to compare the models, three conditions were checked: independence, normality and heteroscedasticity. If all three conditions were satisfied, the use of a parametric test was appropriate (García et al., 2010). This study fulfilled the independence condition since each semi-supervised learning

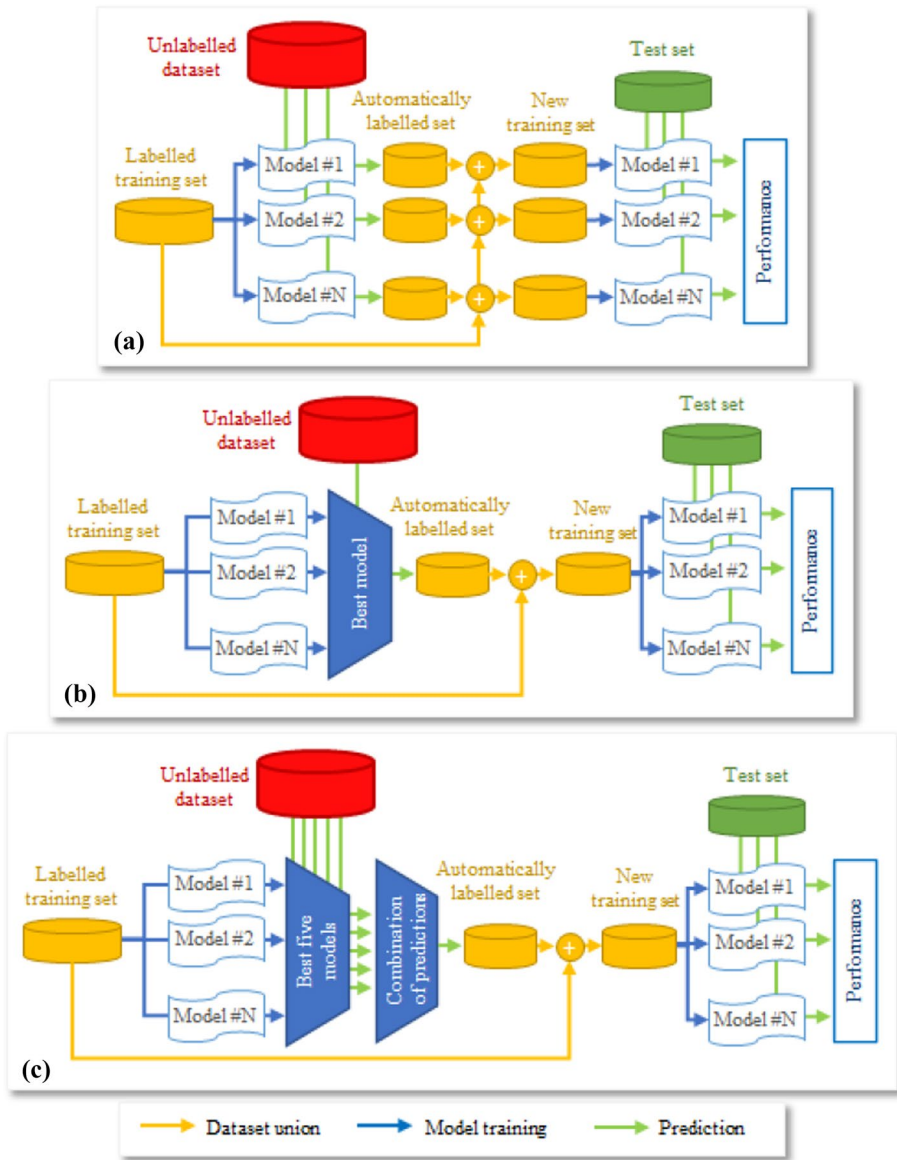


Fig. 3 Schemes of the semi-supervised approaches presented in this analysis: **a** pseudolabeling, **b** distillation and **c** model distillation. Yellow, blue and green arrows refer to the processes of dataset union between manual and automatically labeled datasets, model training on the corresponding training set and prediction of the input images with the model crossed by the arrow, respectively. The models enumerated from 1 to N represent the architectures and backbones of Table 1

approach was independent of the others. Normality was checked by the Shapiro–Wilk test (Shapiro & Wilk, 1965), where the null hypothesis consisted of the normal of the data. Finally, the heteroscedasticity was checked by the Levene test (Levene, 1960), where the null hypothesis was that the results are heteroscedastic.

Since more than two training approaches were compared, an ANOVA test was used when parametric conditions were fulfilled, while a Friedman test was used otherwise (Sheskin, 2011). In both cases, the null hypothesis was that all the training approaches had the same performance. After checking which method was statistically better than the others, a post-hoc procedure was employed to address the multiple hypothesis testing among the different approaches. A Holm post-hoc procedure (Holm, 1979), in the non-parametric case, or a Bonferroni-Dunn post-hoc procedure (Sheskin, 2011), in the parametric case, was used for detecting the significance of the multiple comparisons (Garcia et al., 2010; Sheskin, 2011) and whether the p-values should be corrected and adjusted. The level of confidence of the experimental analysis was set to 0.05. In addition, the size effect was measured using Cohen's d (Cohen, 1969) and Eta Squared (Cohen, 1973).

Results and discussion

The performance of the trained networks (both by applying and without applying the semi-supervised learning methods) was first evaluated considering an independent test set of 25 images. Performance was first assessed using the BLDO labels to compare the results with those in Marani et al. (2021), where several classification networks (namely, AlexNet, GoogleNet, VGG16 and VGG19) were implemented to construct probability maps from image patches generated using a sliding window. Then, the best models were trained and tested using the OSO labels to show the influence of manual annotation on the segmentation results. Finally, the time performance on inference time of the different architectures was analyzed.

Evaluation of the semi-supervised learning methods

All but two deep segmentation networks trained without semi-supervised learning methods, see Table 2, outperformed the approach presented in Marani et al. (2021). Namely, the total MSA (average of the five MSA_c values) of the best segmentation model improved by more than 15%, and the bunch MSA more than 5%. It is worth mentioning that the approach presented in Marani et al. (2021) was aimed to help only the segmentation of the bunch class. For this reason, the improvement in the segmentation of the bunch class was lower than the one of the other classes, which was much more considerable.

If the segmentation networks were compared, there were four networks (DeepLabV3+-ResNext50, Manet-EfficientnetB3, Manet-Resnet50 and Unet++-ResNet50) with a total MSA of over 84%. Among them, the DeepLabV3+-ResNext50 showed better segmentation accuracy than the other networks. With the focus on the bunch class, the DeepLabV3+-ResNext50 and the Manet-EfficientnetB3 networks shined before the others achieving an MSA over 85% for that class. The Pan-Resnet50 model produced the best segmentation of the leaves, while the Unet++-ResNet50 model outperformed the others for the pole class and the Manet-Resnet50 model for the wood class. This illustrates the importance of testing different architectures since they focus on various aspects of the images. Therefore, they can be employed with different aims. For instance, if the final objective is measuring the production of grape bunches, DeepLabV3+-ResNext50 or Manet-EfficientnetB3 models should be used since they provided the best accuracy for the bunch class. In contrast, if this segmentation aims at trimming, Manet-Resnet50 model should be used since it offered the best accuracy for the wood class.

Table 2 Mean segmentation accuracy (percentage) computed on test images of the deep learning models trained by the manually labeled dataset

Network	Background	Leaves	Pole	Bunch	Wood	Total
AlexNet	76.91	74.32	54.86	74.80	66.77	69.52
GoogleNet	69.61	72.80	51.55	74.41	59.09	65.49
VGG16	76.71	74.82	76.46	73.73	47.13	69.77
VGG19	80.53	68.99	76.05	80.58	36.97	68.63
Bisenet-ResNet18	86.26	73.27	82.43	81.99	82.75	81.60
Bisenet-ResNet34	83.46	77.55	83.80	83.54	84.01	82.93
CgNet	84.80	73.46	82.03	82.10	82.34	81.33
ContextNet	81.18	75.85	82.48	82.21	82.60	81.42
DeepLabV3+-EfficientnetB3	82.92	79.04	84.15	83.75	84.46	83.31
DeepLabV3+-ResNext50	85.23	80.67	85.31	85.09	85.85	84.78
DeepLabV3+-ResNet50	79.59	68.30	78.25	78.02	78.54	77.13
DenseApp	87.21	67.29	79.19	78.84	79.72	78.59
FPENet	15.05	39.10	28.87	28.28	28.63	28.49
HRNet	90.42	73.93	83.37	83.24	83.93	83.08
LedNet	82.89	69.93	80.35	80.13	80.28	79.20
Manet-EfficientnetB3	84.63	80.19	85.36	85.69	85.45	84.69
Manet-Resnest50	84.48	79.74	85.01	84.67	86.02	84.42
Manet-Resnet50	81.07	78.98	84.85	84.51	85.17	83.63
OCNet	82.67	76.69	82.65	82.59	83.45	82.06
Pan-EfficientnetB3	81.33	76.45	82.28	82.11	82.51	81.44
Pan-Resnet50	79.95	82.09	84.45	84.04	84.72	83.58
Unet-EfficientnetB3	84.94	70.93	82.27	82.04	82.62	81.09
Unet-Resnest50	86.93	66.02	80.18	80.04	80.64	79.15
Unet-Resnet50	87.91	73.20	82.68	82.29	83.76	82.20
Unet+-EfficientnetB3	87.18	76.72	83.83	83.85	84.59	83.49
Unet+-ResNest50	20.76	71.58	50.68	50.61	50.85	50.03
Unet+-ResNet50	85.40	78.92	85.57	84.60	85.22	84.35

Results in bold are the best

In addition to the raw numbers, several conclusions can be drawn by observing the segmentations of the best model for each class in Fig. 4. For the same image, even if all the models achieved a mean bunch segmentation accuracy of over 80%, only the Manet-EfficientnetB3 model could detect three of the four grape bunches. In addition, some leaves partially occluded the last bunch, making segmentation difficult since that region was segmented as either background or leaves by all the models.

The impact of the different semi-supervised learning methods for the networks studied is provided in Table 3—the results of the semi-supervised methods for each class are in the appendix. At the same time, Fig. 5 shows the effects of applying these approaches on the segmentation mask output of the DeepLabV3+-ResNext50, which produced the best total MSA with plain training. From Fig. 5, it can be noticed that the segmentations made by using the semi-supervised learning methods were less noisy than those produced by the original models. This happens because the semi-supervised methods helped to smooth the predictions. It is also worth mentioning that training using semi-supervised learning

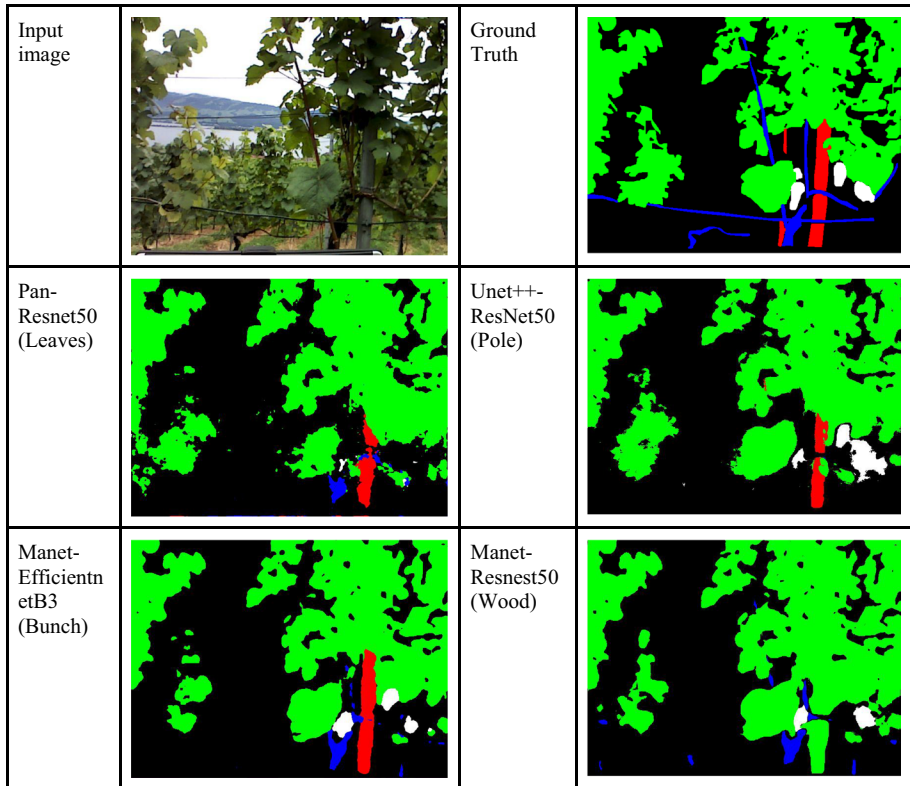


Fig. 4 Example of the segmentation results using the best model for each class

methods could help detect objects, like grape bunches in the pseudolabeling approach of Fig. 5, that were not previously seen by the models trained only with the manually annotated data.

With more details, the pseudolabeling approach produced a mean improvement of 5.62% (with a standard deviation of 13.04%). Only four networks got worse results using this training approach while, in some cases, namely for the FPENet model in Fig. 6, the improvement was over 55%. In Fig. 6, grape bunches and other objects that were not segmented with the initial FPENet model were correctly detected using the FPENet version trained with the pseudolabeling approach. Similarly, the distillation method produced a mean improvement of 6.01% (with a standard deviation of 12.91%), with only two networks having worse results. Finally, the model distillation method also considerably improved the performance of the models (a mean of 5.80% with a standard deviation of 12.90%). However, this improvement was slightly lower than the distillation approach.

As stated before, a statistical analysis was performed to determine significant differences among the training procedures. Since the normality condition was not fulfilled (Shapiro–Wilk’s test $W=0.313172$; $p=0.000000$), Friedman’s non-parametric test was employed to compare the training procedures. Friedman’s test performed a ranking of the training procedures under comparison (see Table 4), assuming as null hypothesis that all the models have the same performance. In this case, significant differences arised

Table 3 Total mean segmentation accuracy (percentage) from applying the different semi-supervised learning procedures to label the testing images

Network	Plain training	Pseudolabeling	Distillation	Model distillation
Bisnet-ResNet18	81.60	83.91	84.00	81.60
Bisnet-ResNet34	82.93	84.10	83.78	83.63
CgNet	81.33	82.90	83.54	83.14
ContextNet	81.42	78.59	83.44	83.24
DeepLabV3+-EfficientnetB3	83.31	84.82	84.96	84.82
DeepLabV3+-ResNext50	84.78	85.45	85.45	85.86
DeepLabV3+-ResNet50	77.13	85.54	85.49	85.49
DenseApp	78.59	83.64	83.47	82.89
FPENet	28.49	83.52	83.68	83.28
HRNet	83.08	84.89	85.07	85.19
LedNet	79.20	83.70	84.72	84.65
Manet-EfficientnetB3	84.69	85.54	85.54	85.54
Manet-Resnest50	84.42	83.75	84.57	83.75
Manet-Resnet50	83.63	83.43	82.85	83.43
OCNet	82.06	83.05	82.46	82.43
Pan-EfficientnetB3	83.58	85.39	83.39	83.42
Pan-Resnet50	81.44	81.97	83.57	83.62
UNet-EfficientnetB3	81.09	83.69	84.87	83.69
UNet-Resnest50	79.15	85.37	85.37	85.37
UNet-Resnet50	82.20	85.06	85.06	85.06
Unet+-EfficientnetB3	83.49	82.12	84.45	84.45
Unet+-ResNest50	50.03	85.26	85.26	85.26
Unet+-ResNet50	84.35	85.66	85.37	85.66

The result in bold is the best.

($F = 15.66$; $p < 8.48e^{-8}$) with a large size effect eta squared 0.13. The distillation method produced the best models. Moreover, looking at the standard deviation values of Table 4, the performance variability produced by the distillation approach is considerably reduced compared with plain training. Consequently, models can be trained more efficiently but can lead to poor results if only manually annotated data is used.

Table 5 shows the results of the application of the Holm algorithm to compare the control training procedure (winner, based on distillation) with all the other training approaches, adjusting the p-value. Results proved significant differences between the semi-supervised learning procedures and the plain training approach, while all the semi-supervised learning methods produced the same outcomes. The size effect was also taken into account using Cohen's d, and, as shown in Table 5, it is medium or large when the winning approach was compared with the rest of the models.

In summary, semi-supervised learning methods provided a considerable boost to all segmentation models without requiring the annotation of additional images. Providing precise annotations was a time-consuming task, and, therefore, reducing the annotation load could help the adoption of deep learning methods. However, deep learning models can only learn what is provided in the annotations. For segmentation tasks in agriculture, several small objects are annotated as background, making unfeasible their automatic segmentation, even



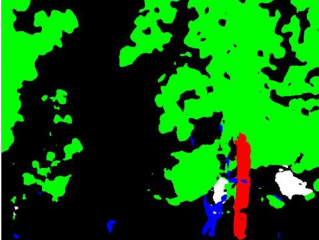
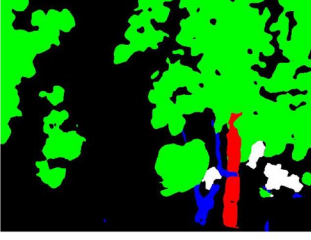
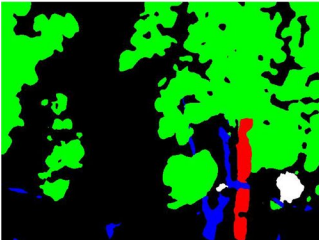
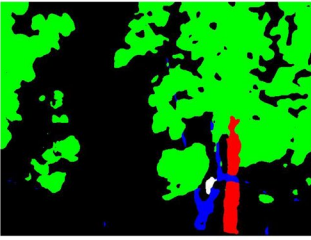
Input image		Ground truth	
Plain		Pseudo labeling	
Distillation		Model distillation	

Fig. 5 Example of the segmentation results using DeepLab v3+-ResNext50 with the four training strategies


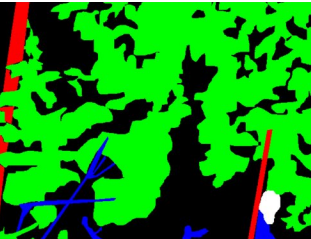
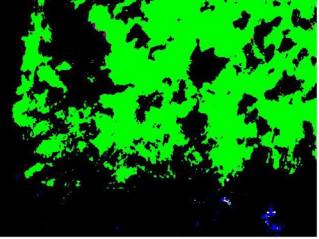

Input image		Ground truth	
FPENet Plain		FPENet Pseudo labeling	

Fig. 6 Example of the segmentation results using the FPENet with plain training and using pseudolabeling

Table 4 Friedman’s test for the mean Total MSA of the training methods

Training technique	Mean Total MSA (std)	Friedman’s test average ranking
Plain training	78.37 (12.63)	1.2246
Pseudolabelling	83.97 (1.58)	2.7518
Distillation	84.36 (0.91)	3.2808
Model distillation	84.15 (1.14)	2.7427

Table 5 Adjusted p-values with Holm and Cohen’s d

Training technique	Z value	p-value	Adjusted p-value	Cohen’s d
Pseudolabelling	0.992945	0.320737	0.625041	0.2966
Model distillation	1.00995	0.312521	0.625041	0.2011
Plain training	3.85956	0.00011359	0.000340771	0.6570

Control technique: Distillation

applying semi-supervised learning methods. This could be solved by a more fine-grained annotation, implementing object-segmentation-oriented (OSO) labels, as show in the next section.

Evaluations with OSO labels

As described in the “Input dataset” subsection, a different annotation scheme was followed to produce more refined labels suitable for object segmentation models (OSO labels). These labels were used to train the same segmentation models of Table 1, following the plain training approach. The new results of the different architectures trained with the OSO labels are shown in Table 6.

Several models achieved a total MSA of over 85%, including DeepLabV3+-ResNet50, Pan-Resnet50, HRNet and all the versions of the Unet and Unet++ architectures. The best overall model was HRNet, with a total MSA of 85.91%. This model also obtained the best accuracy for the leaves, pole and bunch classes. In contrast, the best models for segmenting wood and the background were based on the Unet++ architecture. The outstanding results of the HRNet model were due to the design of its architecture, which aggregated the output representations at four different resolutions, thus allowing models to provide a precise segmentation of objects with different scales.

Segmentation maps in Fig. 7 help draw additional conclusions about the models trained with the BLDO and OSO labels. For the same image, the best overall model trained with the BLDO labels (DeepLabV3+-ResNext50 using model distillation) and the best model trained with the OSO labels (HRNet using plain training) could both segment grape bunches and leaves. However, the segmentation of smaller objects, such as small wooden fragments, was much better when models were trained with OSO labels. In contrast, BLDO labels were not accurate enough to train a model of such small objects, even using any semi-supervised learning approach.

Whether it is better to produce a dataset with a coarse annotation that is later combined with semi-supervised learning methods or a dataset with a fine-grained annotation depend on the final aim of the trained models. Production monitoring or vegetation

Table 6 Mean segmentation accuracy (percentage) computed on test images of the deep learning models trained on the dataset of OSO labels

Network	Background	Leaves	Pole	Bunch	Wood	Total
Bisenet-ResNet18	74.45	71.24	79.37	79.47	81.92	78.32
Bisenet-ResNet34	82.01	78.27	83.72	83.66	86.11	83.33
CgNet	83.01	77.46	83.76	83.87	85.47	83.26
ContextNet	78.05	78.55	82.97	83.20	85.21	82.38
DeepLabV3+-EfficientnetB3	87.35	77.43	84.52	84.48	85.85	84.21
DeepLabV3+-ResNext50	84.11	81.1	85.25	85.13	86.65	84.85
DeepLabV3+-ResNet50	86.66	79.47	85.31	85.31	86.65	85.02
DenseApp	81.53	75.47	82.05	82.23	84.39	81.70
FPENet	66.56	69.72	74.27	74.34	76.55	73.29
HRNet	85.41	82.56	86.21	86.31	87.25	85.91
LedNet	78.47	76.84	82.89	82.55	85.02	81.97
Manet-EfficientnetB3	84.63	80.19	85.36	85.69	85.45	84.69
Manet-Resnest50	80.33	55.19	71.54	71.07	72.88	70.59
Manet-Resnet50	82.94	76.84	84.03	83.91	85.95	83.35
OCNet	82.96	73.12	81.49	81.55	84.02	81.15
Pan-EfficientnetB3	82.21	81.40	85.03	85.12	86.30	84.53
Pan-Resnet50	86.02	79.73	85.39	85.56	86.82	85.09
Unet-EfficientnetB3	86.89	79.38	85.39	85.51	86.78	85.12
Unet-Resnest50	88.06	79.72	85.75	85.82	87.14	85.56
Unet-Resnet50	86.90	80.31	85.57	85.67	86.78	85.35
Unet+-EfficientnetB3	87.29	79.60	85.71	85.81	87.30	85.49
Unet+-ResNest50	88.28	79.99	86.03	86.03	87.25	85.78
Unet+-ResNet50	87.27	80.88	85.84	85.81	86.73	85.56

Results in bold are the best

indices estimation require the segmentation of the main objects of the images (bunches and leaves), achievable even with coarse datasets carrying information about their appearance. However, tasks like trimming or robot harvesting require more precise segmentation to interact with the environment appropriately. Here, it was mandatory to invest more time and effort in producing a fine-grained annotation of the images.

Time inference performance

This comparative study ended with the analysis of the inference time of the models since producing segmentation in a reasonable time is as crucial as obtaining precise results. This will enable their actual implementation for accurate yield monitoring and robot harvesting in almost real-time. The inference times of each model using an Nvidia RTX 2080 Ti GPU and an Intel(R) Core(TM) i7-4790 CPU @ 3.60 GHz are shown in Fig. 8. It is worth noticing that the inference time was independent of the training method or the dataset used to construct the models as it only depends on the selected architecture. The DeepLabV3+-ResNext50 model, which obtained the best accuracy with BLDO labels, could process 100 images in 26.1 ms using a GPU and 315 with a CPU; whereas the HRNet model, which obtained the best accuracy with the OSO labels, processed 100 images in 26.3 ms

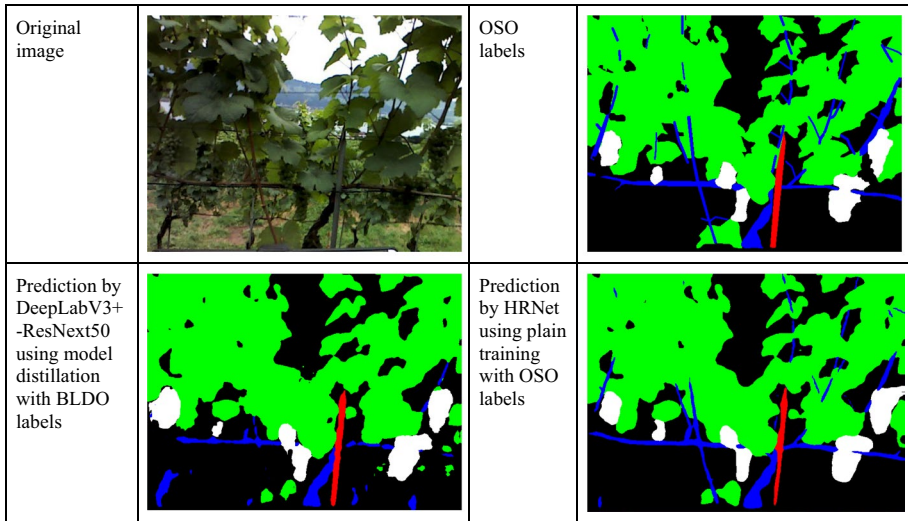


Fig. 7 Comparison of the results obtained with the best model trained on the BLDO labels (DeepLabV3+-ResNext50 using model distillation) and the best model (HRNet) trained on the refined OSO labels

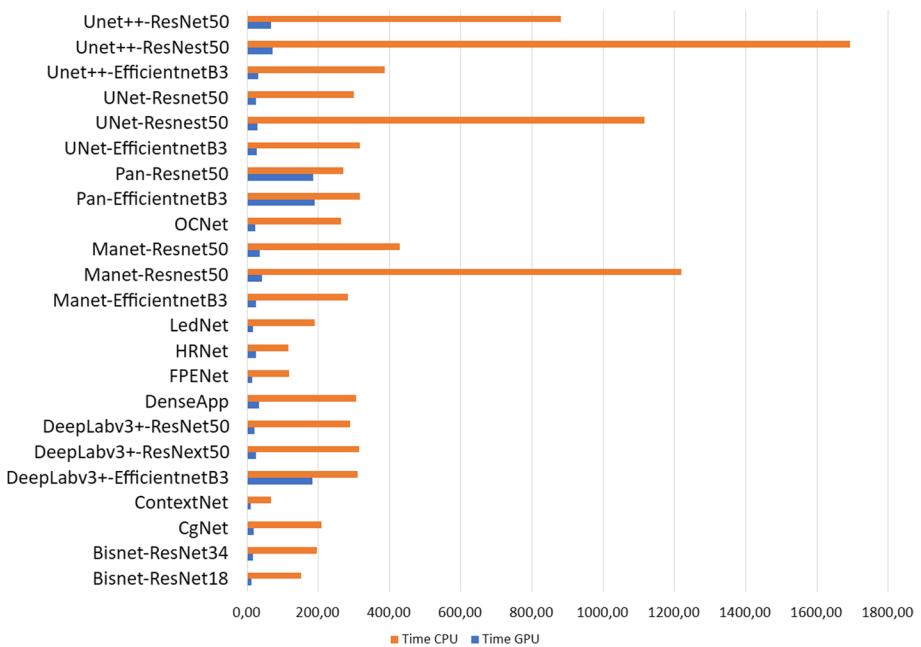


Fig. 8 Inference time (in milliseconds) for 100 images of each segmentation model using CPU and GPU

using a GPU and 118 ms with a CPU. The best model at inference time was the ContextNet model, which segmented 100 images in 11.6 ms using a GPU and 68.9 ms using a CPU. This model also provided the best trade-off between accuracy and inference time.

Therefore, ContextNet would be the preferred model to be implemented in-field for real-time processing.

Future work

Future works will address the use of infrared and depth streams returned by the proposed cameras as input of the proposed models or as the object of investigation for accurate yield monitoring. Moreover, hardware-aware models or quantization methods will be explored to integrate the segmentation models in low-cost devices used in-field. The more significant amount of feeding information will lead to better segmentation results and even to the direct regression of crop productivity.

Conclusions

Analyzing natural images captured by moving robotic platforms is a key point for yield monitoring at the plant level. Its actual implementation requires low-cost sensors, such as RGB-D cameras, able to provide detailed information about both appearance and volume of the targets, e.g. the whole plants or single fruits. As a first step in using these data, reliable software methods are mandatory to process low-quality color images and give helpful knowledge to the farmers.

In the scenario of viticulture, this paper presented:

- several deep learning architectures for the segmentation of natural color images acquired in vineyards by the Intel Realsense R200 stereo camera;
- three semi-supervised approaches to improve segmentation accuracy by taking advantage of a set of unlabeled images, thus avoiding the need for a large dataset of labeled images, whose annotation can be time-demanding; and
- a comprehensive discussion on the need for high-detailed manual annotation for improving environmental awareness.

Results showed that the DeepLabV3+ -ResNext50 model, trained by the set of labeled images, achieves the best MSA of 84.78% (average of the MSAs of all the target classes), whereas the Manet-EfficientnetB3 model reaches the MSA of the bunch class (85.69%) under the same training conditions. On average, the application of semi-supervised learning methods boosted MSAs between 5.62 and 6.01%. In particular, the model distillation semi-supervised approach improved the total MSA of the DeepLabV3+ -ResNext50 model to 85.86%. However, other architectures, such as the FPENet, benefit more than 55% in MSA from the semi-supervised approaches, which de facto enabled the creation of appropriate models. Finally, also time-efficiency was investigated, proving that the ContextNet model almost halved the inference time of the DeepLabV3+ -ResNext50 model at the expense of a slight worsening of the total MSA, which in the case of the distillation semi-supervised learning procedure, reaches 83.44%.

A final comparison of models trained with two label sets, oriented at bunch/leaves detection and object segmentation, was presented to show the effect of manual annotation. Specifically, coarse labels can be efficiently used to model objects of large sizes, such as grape bunches or leaf clusters, making them suitable for production monitoring and

vegetation indices estimation. In contrast, those applications requiring the exact environmental awareness, such as robotic harvesting or trimming, must use more detailed labels to create exhaustive segmentation models.

Appendix

This appendix reports the mean segmentation accuracies for each class varying the semi-supervised learning methods, extending the results of Table 3. Tables 7, 8 and 9 refer to pseudolabeling, distillation and model distillation, respectively.

Table 7 Mean segmentation accuracy computed on test images of the deep learning models trained by using the pseudolabeling semi-supervised method

Network	Background	Leaves	Pole	Bunch	Wood	Total
Bisenet-ResNet18	83.59	79.92	84.5	84.32	85.09	83.91
Bisenet-ResNet34	85.29	79.09	84.58	84.57	85.18	84.1
CgNet	87.83	75.02	83.37	83.28	83.85	82.9
ContextNet	87.21	67.29	79.19	78.84	79.72	78.59
DeepLabV3+-EfficientnetB3	82.29	83.11	85.37	85.22	85.85	84.82
DeepLabV3+-ResNext50	86.43	81.74	85.82	85.61	86.31	85.45
DeepLabV3+-ResNet50	83.36	83.79	85.93	85.92	86.62	85.54
DenseApp	84.93	78.45	84.13	84.32	84.54	83.64
FPENet	81.25	79.53	84.6	84.37	84.7	83.52
HRNet	87.34	79.94	85.16	85.26	85.64	84.89
LedNet	85.85	77.83	84.18	84.33	84.62	83.7
Manet-EfficientnetB3	87.56	80.66	85.94	85.89	86.35	85.54
Manet-Resnest50	83.73	77.51	84.54	84.74	85.35	83.75
Manet-Resnet50	88.93	75.74	83.84	83.71	84.21	83.43
OCNet	82.85	79.32	83.59	83.43	84.1	83.05
Pan-EfficientnetB3	84.9	82.96	85.92	85.6	86.1	85.39
Pan-Resnet50	85.84	74.12	82.78	82.41	83.08	81.97
Unet-EfficientnetB3	89.29	75.73	84.16	83.92	84.58	83.69
Unet-Resnest50	85.89	82.36	85.63	85.57	86.13	85.37
Unet-Resnet50	82.5	83.08	85.43	85.23	86.68	85.06
Unet+-EfficientnetB3	79.79	77.12	83.31	83.09	83.68	82.12
Unet+-ResNest50	86.01	81.36	85.69	85.54	86.18	85.26
Unet+-ResNet50	87.15	82.01	85.87	85.8	86.43	85.66

Results in bold are the best

Table 8 Mean segmentation accuracy computed on test images of the deep learning models trained by using the distillation semi-supervised method

Network	Background	Leaves	Pole	Bunch	Wood	Total
Bisenet-ResNet18	82.36	81.61	84.55	84.25	85.1	84
Bisenet-ResNet34	82.8	80.73	84.34	84.08	84.84	83.78
CgNet	87.13	76.91	83.97	93.92	84.48	83.54
ContextNet	82.59	80.03	83.92	84.04	84.41	83.44
DeepLabV3+-EfficientnetB3	82.45	83.21	85.52	85.36	86.02	84.96
DeepLabV3+-ResNext50	86.43	81.74	85.82	85.61	86.31	85.45
DeepLabV3+-ResNet50	83.52	83.16	86.23	85.78	86.51	85.49
DenseApp	85.44	78.54	83.91	83.76	84.32	83.47
FPENet	81.5	79.69	84.79	84.51	84.82	83.68
HRNet	89.01	78.22	85.32	85.47	85.93	85.07
LedNet	85.63	80.56	85.15	85.07	85.64	84.72
Manet-EfficientnetB3	87.56	80.66	85.94	85.89	86.35	85.54
Manet-Resnest50	84.62	80.47	85.66	84.78	85.33	84.57
Manet-Resnet50	78.79	79.33	84.05	83.82	84.42	82.85
OCNet	83.66	77.58	82.86	83	83.49	82.46
Pan-EfficientnetB3	80.62	80.72	84.18	84.07	84.63	83.39
Pan-Resnet50	82.69	79.34	84.31	84.37	84.75	83.57
Unet-EfficientnetB3	88.58	78.37	85.33	85.14	85.78	84.87
Unet-Resnest50	85.89	82.36	85.63	85.57	86.13	85.37
Unet-Resnet50	82.5	83.08	85.43	85.23	86.68	85.06
Unet+-EfficientnetB3	88.82	77.83	84.77	84.76	85.22	84.45
Unet+-ResNest50	86.01	81.36	85.69	85.54	86.18	85.26
Unet+-ResNet50	85.89	82.33	85.63	85.57	86.13	85.37

Results in bold are the best

Table 9 Mean segmentation accuracy computed on test images of the deep learning models trained by using the model distillation semi-supervised method

Network	Background	Leaves	Pole	Bunch	Wood	Total
Bisenet-ResNet18	86.26	73.27	82.43	81.99	82.75	81.6
Bisenet-ResNet34	81.77	81.14	84.23	84.09	84.72	83.63
CgNet	87.3	75.8	83.62	83.54	84.11	83.14
ContextNet	82.62	79.77	83.68	83.87	84.2	83.24
DeepLabV3+-EfficientnetB3	82.75	82.69	85.41	85.19	85.88	84.82
DeepLabV3+-ResNext50	84.88	83.24	86.3	86.3	86.72	85.86
DeepLabV3+-ResNet50	84.06	83.17	85.9	85.88	86.52	85.49
DenseApp	84.38	78.22	83.27	83.23	83.85	82.89
FPENet	78.61	81.2	84.46	84.2	84.46	83.28
HRNet	88.94	78.51	85.43	85.61	86.06	85.19
LedNet	84.35	81.28	85.09	85.08	85.26	84.65
Manet-EfficientnetB3	87.56	80.66	85.94	85.89	96.35	85.54
Manet-Resnest50	83.73	77.51	84.54	84.74	85.35	83.75
Manet-Resnet50	88.93	75.74	83.84	83.71	84.21	83.43
OCNet	81.88	79.6	82.8	82.64	83.47	82.43
Pan-EfficientnetB3	82.6	79.3	84.14	84.11	84.49	83.42
Pan-Resnet50	80.78	80.88	84.49	84.29	84.79	83.62
Unet-EfficientnetB3	89.29	75.73	84.16	83.92	84.58	83.69
Unet-Resnest50	85.89	82.36	85.63	85.57	86.13	85.37
Unet-Resnet50	82.5	83.08	85.43	85.23	86.68	85.06
Unet+-EfficientnetB3	88.82	77.83	84.77	84.76	85.22	84.45
Unet+-ResNest50	86.01	81.36	85.69	85.54	86.18	85.26
Unet+-ResNet50	87.15	82.01	85.87	85.8	86.43	85.66

Results in bold are the best

Funding This work was partially supported by Ministerio de Ciencia e Innovación [PID2020-115225RB-I00/AEI/<https://doi.org/10.13039/501100011033>]. Ángela Casado-García has a FPI grant from Community of La Rioja 2020. The financial support of the following grants is also acknowledged: Agricultural Interoperability and Analysis System (ATLAS), European Union's Horizon 2020 research and innovation programme (Grant No. 857125), Multimodal Sensing for Individual Plant Phenotyping in agriculture robotics (ANTONIO), ICT-AGRI-FOOD COFUND (Grant No. 41946), E-CROPS - Tecnologie per l'Agricoltura Digitale Sostenibile, PON Ricerca e Innovazione 2014–2020 (Grant No. ARS01_01136). The authors are also grateful to M. Attolico, G. Bono, S. Rilling, P. Frölich and M. Nielsen for their support in performing the experiments and gathering the data. The sole responsibility for the content of this publication lies with the authors. It does not necessarily represent the opinion of the European Union. Neither the EASME nor the European Commission is responsible for any use that may be made of the information contained therein.

Data availability The data is available at <https://github.com/ispstiima/S3CavVineyardDataset>.

Code availability All the code developed for this work is available at <https://github.com/ancasag/GrapeBunchSegmentation>.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Adão, T., Hruška, J., Pádua, L., Bessa, J., Peres, E., Morais, R., et al. (2017). Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry. *Remote Sensing*, 9(11), 1110. <https://doi.org/10.3390/rs9111110>
- Afonso, M., Fonteijn, H., Fiorentin, F. S., Lensink, D., Mooij, M., Faber, N., et al. (2020). Tomato fruit detection and counting in greenhouses using deep learning. *Frontiers in Plant Science*, 11, 1759. <https://doi.org/10.3389/fpls.2020.571299>
- Barnea, E., Mairon, R., & Ben-Shahar, O. (2016). Colour-agnostic shape-based 3D fruit detection for crop harvesting robots. *Biosystems Engineering*, 146, 57–70. <https://doi.org/10.1016/j.biosystemseng.2016.01.013>
- Barriguinha, A., de Castro Neto, M., & Gil, A. (2021). Vineyard yield estimation, prediction, and forecasting: A systematic literature review. *Agronomy*, 11(9), 1789. <https://doi.org/10.3390/agronomy11091789>
- Behroozi-Khazaei, N., & Maleki, M. R. (2017). A robust algorithm based on color features for grape cluster segmentation. *Computers and Electronics in Agriculture*, 142, 41–49. <https://doi.org/10.1016/j.compag.2017.08.025>
- Berenstein, R., Shahar, O. B., Shapiro, A., & Edan, Y. (2010). Grape clusters and foliage detection algorithms for autonomous selective vineyard sprayer. *Intelligent Service Robotics*, 3(4), 233–243. <https://doi.org/10.1007/s11370-010-0078-z>
- Bosilj, P., Aptoula, E., Duckett, T., & Cielniak, G. (2020). Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture. *Journal of Field Robotics*, 37(1), 7–19. <https://doi.org/10.1002/rob.21869>
- Bucilua C., Caruana, R., & Niculescu-Mizil, A. (2006). Model compression: making big, slow models practical. In *Proceedings of the 12th International conference on knowledge discovery and data mining (KDD'06)* (pp. 535–541). New York, USA: Association for Computing Machinery. <https://doi.org/10.1145/1150402.1150464>.
- Chebrolu, N., Lottes, P., Schaefer, A., Winterhalter, W., Burgard, W., & Stachniss, C. (2017). Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *The International Journal of Robotics Research*, 36(10), 1045–1052. <https://doi.org/10.1177/0278364917720510>
- Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 801–818). https://doi.org/10.1007/978-3-030-01234-2_49
- Cohen, J. (1969). *Statistical power analysis for the behavioral sciences*. Academic Press.
- Cohen, J. (1973). Eta-squared and partial eta-squared in fixed factor ANOVA designs. *Educational and Psychological Measurement*, 33, 107–112. <https://doi.org/10.1177/001316447303300111>
- Das, J., Cross, G., Qu, C., Makineni, A., Tokekar, P., Mulgaonkar, Y., et al. (2015, August). Devices, systems, and methods for automated monitoring enabling precision agriculture. In *2015 IEEE international conference on automation science and engineering (CASE)* (pp. 462–469). IEEE. <https://doi.org/10.1109/CoASE.2015.7294123>

- Dyson, J., Mancini, A., Frontoni, E., & Zingaretti, P. (2019). Deep learning for soil and crop segmentation from remotely sensed data. *Remote Sensing*, 11(16), 1859. <https://doi.org/10.3390/rs11161859>
- Fu, L., Gao, F., Wu, J., Li, R., Karkee, M., & Zhang, Q. (2020a). Application of consumer RGB-D cameras for fruit detection and localization in field: A critical review. *Computers and Electronics in Agriculture*, 177, 105687.
- Fu, L., Majeed, Y., Zhang, X., Karkee, M., & Zhang, Q. (2020b). Faster R-CNN-based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting. *Biosystems Engineering*, 197, 245–256. <https://doi.org/10.1016/j.biosystemseng.2020.07.007>
- Gao, F., Fu, L., Zhang, X., Majeed, Y., Li, R., Karkee, M., et al. (2020). Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN. *Computers and Electronics in Agriculture*, 176, 105634. <https://doi.org/10.1016/j.compag.2020.105634>
- García, S., Fernández, A., Luengo, J., & Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180(10), 2044–2064. <https://doi.org/10.1016/j.ins.2009.12.010>
- Gongal, A., Silwal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2016). Apple crop-load estimation with over-the-row machine vision system. *Computers and Electronics in Agriculture*, 120, 26–35. <https://doi.org/10.1016/j.compag.2015.10.022>
- Guo, W., Zheng, B., Potgieter, A. B., Diot, J., Watanabe, K., Noshita, K., et al. (2018). Aerial imagery analysis—quantifying appearance and number of sorghum heads for applications in breeding and agronomy. *Frontiers in Plant Science*, 9, 1544. <https://doi.org/10.3389/fpls.2018.01544>
- Heras, J., Marani, R., & Milella, A. (2021). Semi-supervised semantic segmentation for grape bunch identification in natural images. In J. V. Stafford (Ed.), *Proceedings of the 13th European conference on precision agriculture. Precision Agriculture '21* (pp. 65–84). The Netherlands: Wageningen Academic Publishers, https://doi.org/10.3920/978-90-8686-916-9_39
- Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. Non-peer reviewed preprint at ArXiv preprint [arXiv:1503.02531](https://arxiv.org/abs/1503.02531).
- Holm, O. J. S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6, 65–70. <https://doi.org/10.2307/4615733>
- Howard, J., & Gugger, S. (2020). *Deep learning for coders with fastai & Pytorch*. O'Reilly Media Inc.
- Jiang, P., Chen, Y., Liu, B., He, D., & Liang, C. (2019). Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. *IEEE Access*, 7, 59069–59080. <https://doi.org/10.1109/ACCESS.2019.2914929>
- Kang, H., & Chen, C. (2020). Fruit detection, segmentation and 3D visualisation of environments in apple orchards. *Computers and Electronics in Agriculture*, 171, 105302. <https://doi.org/10.1016/j.compag.2020.105302>
- Kim, J., Kim, S., Ju, C., & Son, H. I. (2019). Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications. *IEEE Access*, 7, 105100–105115. <https://doi.org/10.1109/ACCESS.2019.2932119>
- Knoll, F. J., Czymmek, V., Poczihoski, S., Holtorf, T., & Hussmann, S. (2018). Improving efficiency of organic farming by using a deep learning classification approach. *Computers and Electronics in Agriculture*, 153, 347–356. <https://doi.org/10.1016/j.compag.2018.08.032>
- Kuan, Y. W., Ee, N. O., & Wei, L. S. (2019). Comparative study of intel R200, Kinect v2, and primesense RGB-D sensors performance outdoors. *IEEE Sensors Journal*, 19(19), 8741–8750. <https://doi.org/10.1109/JSEN.2019.2920976>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Lee, D. H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Proceedings ICML workshop: Challenges in representation learning (WREPL)*
- Levene, H. (1960). Contributions to probability and statistics: Essays in honor of harold hotelling, chapter. Robust tests for equality of variances (pp. 278–330). In *Contributions to probability and statistics: Essays in honor of harold hotelling*. Stanford University Press.
- Li, H., Xiong, P., An, J., & Wang, L. (2018). Pyramid attention network for semantic segmentation. In *Proceedings of the 29th British machine vision conference*. Non-peer reviewed preprint at ArXiv preprint [arXiv:1805.10180](https://arxiv.org/abs/1805.10180)
- Li, R., Zheng, S., Duan, C., Zhang, C., Su, J., & Atkinson, P. M. (2020). Multi-attention-network for semantic segmentation of fine resolution remote sensing images. Non-peer reviewed preprint at ArXiv preprint [arXiv:2009.02130](https://arxiv.org/abs/2009.02130)

- Liu, M., & Yin, H. (2019). Feature pyramid encoding network for real-time semantic segmentation. In *Proceedings of the 30th British machine vision conference*. Non-peer reviewed preprint at ArXiv preprint [arXiv:1909.08599](https://arxiv.org/abs/1909.08599)
- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440). New York, USA: IEEE. <https://doi.org/10.1109/CVPR.2015.7298965>.
- Ma, J., Du, K., Zhang, L., Zheng, F., Chu, J., & Sun, Z. (2017). A segmentation method for greenhouse vegetable foliar disease spots images using color information and region growing. *Computers and Electronics in Agriculture*, *142*, 110–117. <https://doi.org/10.1016/j.inpa.2018.08.010>
- Mack, J., Lenz, C., Teutrine, J., & Steinhage, V. (2017). High-precision 3D detection and reconstruction of grapes from laser range data for efficient phenotyping based on supervised learning. *Computers and Electronics in Agriculture*, *135*, 300–311. <https://doi.org/10.1016/j.compag.2017.02.017>
- Majeed, Y., Karkee, M., & Zhang, Q. (2020). Estimating the trajectories of vine cordons in full foliage canopies for automated green shoot thinning in vineyards. *Computers and Electronics in Agriculture*, *176*, 105671. <https://doi.org/10.1016/j.compag.2020.105671>
- Marani, R., Milella, A., Petitti, A., & Reina, G. (2019). Deep learning-based image segmentation for grape bunch detection. In J. V. Stafford (Ed.), *Proceedings of the 12th European conference on Precision agriculture, Precision agriculture'19* (pp. 791–797). Wageningen, The Netherlands: Wageningen Academic Publishers. <https://doi.org/10.3920/978-90-8686-888-9>.
- Marani, R., Milella, A., Petitti, A., & Reina, G. (2021). Deep neural networks for grape bunch segmentation in natural images from a consumer-grade camera. *Precision Agriculture*, *22*(2), 387–413. <https://doi.org/10.1007/s11119-020-09736-0>
- Milella, A., Marani, R., Petitti, A., & Reina, G. (2019). In-field high throughput grapevine phenotyping with a consumer-grade depth camera. *Computers and Electronics in Agriculture*, *156*, 293–306. <https://doi.org/10.1016/j.compag.2018.11.026>
- Milioto, A., Lottes, P., & Stachniss, C. (2018, May). Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs. In *2018 IEEE international conference on robotics and automation (ICRA)* (pp. 2229–2235). IEEE. <https://doi.org/10.1109/ICRA.2018.8460962>.
- Naranjo-Torres, J., Mora, M., Hernández-García, R., Barrientos, R. J., Fredes, C., & Valenzuela, A. (2020). A review of convolutional neural network applied to fruit image processing. *Applied Sciences*, *10*(10), 3443. <https://doi.org/10.3390/app10103443>
- Nguyen, T. T., Vandevoorde, K., Wouters, N., Kayacan, E., De Baerdemaeker, J. G., & Saeys, W. (2016). Detection of red and bicoloured apples on tree with an RGB-D camera. *Biosystems Engineering*, *146*, 33–44. <https://doi.org/10.1016/j.biosystemseng.2016.01.007>
- Osco, L. P., Nogueira, K., Ramos, A. P. M., Pinheiro, M. M. F., Furuya, D. E. G., Gonçalves, W. N., et al. (2021). Semantic segmentation of citrus-orchard using deep neural networks and multispectral UAV-based imagery. *Precision Agriculture*, *22*, 1171–1188. <https://doi.org/10.1007/s11119-020-09777-5>
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Proceedings advances in neural information processing systems 32* (pp. 8024–8035). Red Hook, NY, USA: Curran Associates, Inc.
- Paulus, S., Behmann, J., Mahlein, A. K., Plümer, L., & Kuhlmann, H. (2014). Low-cost 3D systems: Suitable tools for plant phenotyping. *Sensors*, *14*(2), 3001–3018. <https://doi.org/10.3390/s140203001>
- Poudel, P. K. R., Bonde, U., Liwicki, S., & Zach C. (2018). ContextNet: Exploring context and detail for semantic segmentation in real-time. In *Proceedings of the 29th British machine vision conference*. Non-peer reviewed preprint at ArXiv preprint [arXiv:1805.04554](https://arxiv.org/abs/1805.04554).
- Razavian, A. S., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: An astounding baseline for recognition. In *Proceedings IEEE conference on computer vision and pattern recognition workshops (CVPRW'14)* (pp. 512–519). Non-peer reviewed preprint at ArXiv preprint [arXiv:1403.6382](https://arxiv.org/abs/1403.6382).
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. Wells, & A. Frangi (Eds.), *Medical image computing and computer-assisted intervention – MICCAI 2015*. Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-319-24574-4_28.
- Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). Deepfruits: A fruit detection system using deep neural networks. *Sensors*, *16*(8), 1222. <https://doi.org/10.3390/s16081222>
- Saleem, M. H., Potgieter, J., & Arif, K. M. (2021). Automation in agriculture by machine and deep learning techniques: A review of recent developments. *Precision Agriculture*, *22*(6), 2053–2091. <https://doi.org/10.1007/s11119-021-09806-x>

- Santos, T. T., de Souza, L. L., dos Santos, A. A., & Avila, S. (2020). Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. *Computers and Electronics in Agriculture*, 170, 105247. <https://doi.org/10.1016/j.compag.2020.105247>
- Shapiro, S. S., & Wilk, M. B. (1965). An analysis for variance test for normality (complete samples). *Information Sciences*, 180, 2044–2064. <https://doi.org/10.2307/2333709>
- Sheskin, D. (2011). *Handbook of parametric and nonparametric statistical procedures*. CRC Press.
- Song, Z., Zhou, Z., Wang, W., Gao, F., Fu, L., Li, R., et al. (2021). Canopy segmentation and wire reconstruction for kiwifruit robotic harvesting. *Computers and Electronics in Agriculture*, 181, 105933. <https://doi.org/10.1016/j.compag.2020.105933>
- Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., et al. (2019). High-resolution representations for labeling pixels and regions. Non-peer reviewed preprint at Arxiv preprint: 1904.04514.
- Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018, October). A survey on deep transfer learning. In *International conference on artificial neural networks* (pp. 270–279). Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-030-01424-7_27.
- Tao, Y., & Zhou, J. (2017). Automatic apple recognition based on the fusion of color and 3D feature for robotic fruit picking. *Computers and Electronics in Agriculture*, 142, 388–396. <https://doi.org/10.1016/j.compag.2017.09.019>
- Tian, H., Wang, T., Liu, Y., Qiao, X., & Li, Y. (2020a). Computer vision technology in agricultural automation—A review. *Information Processing in Agriculture*, 7(1), 1–19. <https://doi.org/10.1016/j.inpa.2019.09.006>
- Tian, Y., Yang, G., Wang, Z., Li, E., & Liang, Z. (2020b). Instance segmentation of apple flowers using the improved mask R-CNN model. *Biosystems Engineering*, 193, 264–278. <https://doi.org/10.1016/j.biosystemseng.2020.03.008>
- Wang, A., Xu, Y., Wei, X., & Cui, B. (2020a). Semantic segmentation of crop and weed using an encoder-decoder network and image enhancement method under uncontrolled outdoor illumination. *IEEE Access*, 8, 81724–81734. <https://doi.org/10.1109/ACCESS.2020.2991354>
- Wang, Y., Zhou, Q., Liu, J., Xiong, J., Gao, G., Wu, X., et al. (2019). LEDNet: A lightweight encoder-decoder network for real-time semantic segmentation. Non-peer reviewed preprint at ArXiv preprint [arXiv:1905.02423](https://arxiv.org/abs/1905.02423).
- Wang, X. A., Tang, J., & Whitty, M. (2020b). Side-view apple flower mapping using edge-based fully convolutional networks for variable rate chemical thinning. *Computers and Electronics in Agriculture*, 178, 105673. <https://doi.org/10.1016/j.compag.2020.105673>
- Wosner, O., Farjon, G., & Bar-Hillel, A. (2021). Object detection in agricultural contexts: A multiple resolution benchmark and comparison to human. *Computers and Electronics in Agriculture*, 189, 106404. <https://doi.org/10.1016/j.compag.2021.106404>
- Wu, H., Wiesner-Hanks, T., Stewart, E. L., DeChant, C., Kaczmar, N., Gore, M. A., et al. (2019). Autonomous detection of plant disease symptoms directly from aerial imagery. *The Plant Phenome Journal*, 2(1), 1–9. <https://doi.org/10.2135/tppj2019.03.0006>
- Wu, T., Tang, S., Zhang, R., & Zhang, Y. (2018). CGNet: A Light-weight context guided network for semantic segmentation. Non-peer reviewed preprint at Arxiv preprint: 1811.08201.
- Yang, M. D., Tseng, H. H., Hsu, Y. C., & Tsai, H. P. (2020). Semantic segmentation using deep learning with vegetation indices for rice lodging identification in multi-date UAV visible images. *Remote Sensing*, 12(4), 633. <https://doi.org/10.3390/rs12040633>
- Yang, K., Zhong, W., & Li, F. (2020). Leaf segmentation and classification with a complicated background using deep learning. *Agronomy*, 10(11), 1721. <https://doi.org/10.3390/agronomy10111721>
- Yang, M., Yu, K., Zhang, C., Li, Z., & Yang, K. (2018). DenseASPP for semantic segmentation in street scenes. In *2018 IEEE/CVF conference on computer vision and pattern recognition*. <https://doi.org/10.1109/CVPR.2018.00388>.
- Yu, C., Wang, J., Peng C., Gao C., Yu G., & Sang N. (2018) BiSeNet: Bilateral segmentation network for real-time semantic segmentation. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer vision – ECCV 2018. ECCV 2018. Lecture notes in computer science* (vol. 11217). Springer. https://doi.org/10.1007/978-3-030-01261-8_20
- Yuan, Y., & Wang, J. (2018). Ocnet: Object context network for scene parsing. Non-peer reviewed preprint at ArXiv preprint [arXiv:1809.00916](https://arxiv.org/abs/1809.00916).
- Zhang, J., He, L., Karkee, M., Zhang, Q., Zhang, X., & Gao, Z. (2018). Branch detection for apple trees trained in fruiting wall architecture using depth features and Regions-Convolutional Neural Network (R-CNN). *Computers and Electronics in Agriculture*, 155, 386–393. <https://doi.org/10.1016/j.compag.2018.10.029>

- Zhou, J., Zhou, J., Ye, H., Ali, M. L., Nguyen, H. T., & Chen, P. (2020). Classification of soybean leaf wilting due to drought stress using UAV-based imagery. *Computers and Electronics in Agriculture*, 175, 105576. <https://doi.org/10.1016/j.compag.2020.105576>
- Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., & Liang, J. (2018). UNet++: A nested U-net architecture for medical image segmentation. In D. Stoyanov, et al. (Eds.), *Deep learning in medical image analysis and multimodal learning for clinical decision support. DLMIA 2018, ML-CDS 2018*. Lecture notes in computer science (vol. 11045). Springer. https://doi.org/10.1007/978-3-030-00889-5_1.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.