Available online at www.sciencedirect.com

ScienceDirect

Information Processing in Agriculture 00 (2026) 1–21

Procedia Computer Science

Tailored Refinement of Vision-Language Models for Plant Instance Segmentation

Gianmarco Roggiolani[a,*], Cyrill Stachniss[a,b], Jens Behley[a]

[a] Center for Robotics, University of Bonn, Germany
[b] Lamarr Institute for Machine Learning and Artificial Intelligence, Germany

Abstract

Plant phenotyping involves measuring the morphological and physiological traits of plants and is key in agricultural research, breeding, as well as crop management. Detecting single plant instances is the first step to extract plant-level traits and can be achieved via imaging techniques. Most modern visual instance segmentation systems rely on deep learning approaches, which are powerful but usually require a large amount of training data to achieve accurate and robust performance. Our approach enables automatic generation of plant instance labels from RGB images by combining foundation models with geometric techniques, eliminating the need for human annotations. Our method leverages current state-of-the-art vision-language foundation models and domain-specific knowledge to generate training data *without* the need for human annotations. We use our automatically generated labels to enhance the capabilities of learning-based approaches, incorporating our predicted instances as additional input or as labels during training. We evaluate the quality of our generated labels on various datasets and compare to heuristic and deep-learning methods. The experiments demonstrate that our generated labels match or exceed heuristic and learning-based baselines, achieving a max vegetation Intersection-over-Union of 78.7% when used in combination with Grounded SAM 2.1 and a max Panoptic Quality of 67% when used in combination with Florence2 + SAM2 on the PhenoBench dataset. These results show that the combination of general-purpose models with our novel domain-specific post-processing is a viable and scalable solution for plant phenotyping, enabling a broader applicability without the requirement for manual annotations.

Keywords: Plant Phenotyping, Scene Understanding, Instance Segmentation, Unsupervised Learning

1. Introduction

Modern image-analysis tools performing semantic and panoptic segmentation have huge potential to help plant scientists and agricultural researchers extract information about plant growth and phenotypic information. There exist general segmentation systems [1, 2] as well as domain-specific ones, optimized for urban driving scenes [3, 4, 5], industrial inspection tasks [6, 7], or agricultural robots [8, 9, 10].

The instance segmentation problem was originally tackled using heuristic-based techniques, which exploit geometric background knowledge about the domain. One example in agriculture is knowing that most crops are planted in rows. Such knowledge can be useful for specific applications and can be combined with learning-based systems to bootstrap approaches when training data is unavailable or hard to obtain [11, 12].

Brice et al. [13] used regions as base units for images and partitioned the picture using a decision tree. Similarly, Tomita et al. [14] segmented the image in regions with uniform properties using statistical methods. Subsequently

[*] Corresponding author
Email address: groggiol@uni-bonn.de (Gianmarco Roggiolani)

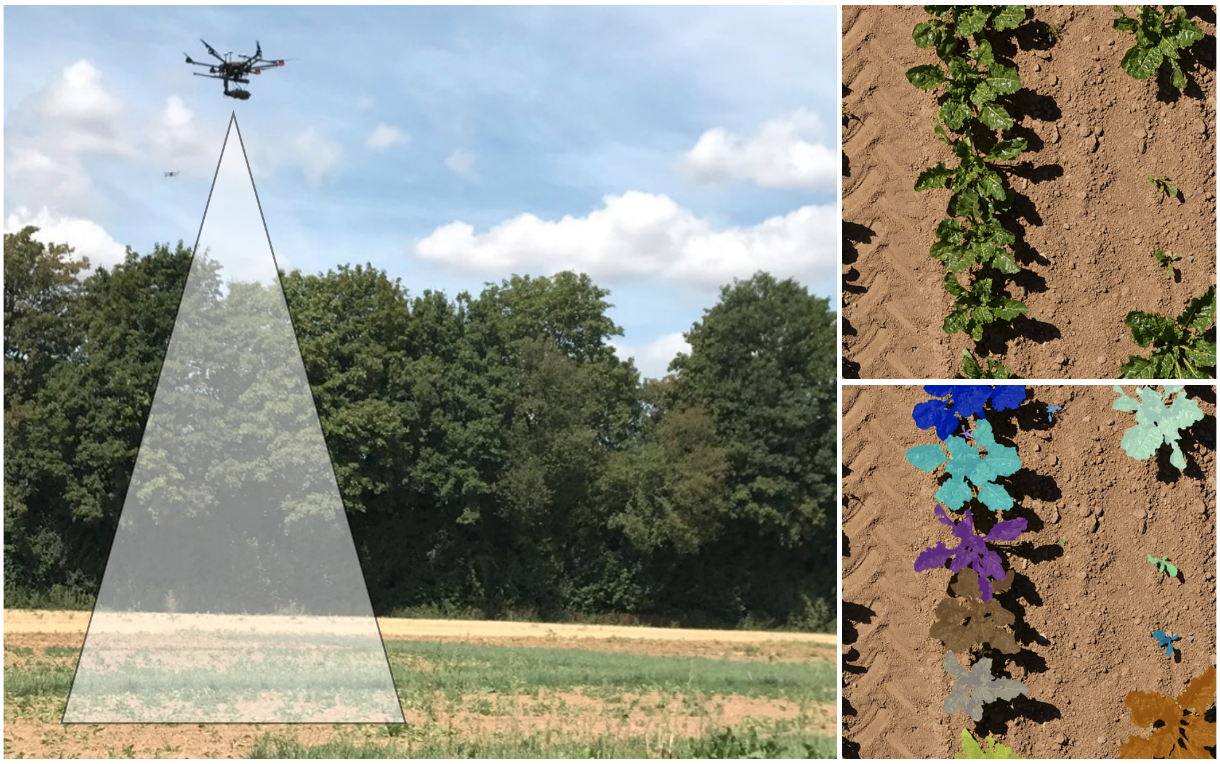


Figure 1: Example of a robotic platform equipped with downwards facing RGB camera to capture images of agricultural fields for monitoring purposes. We can use our approach to perform plant instance segmentation in the captured images and assign a unique identifier to each plant, as shown in the bottom right image, where each color represents a different plant instance.

Wang et al. [15], Pun et al. [16], and Reddi et al. [17] investigated the use of multiple thresholds to capture segments and instances in the images, and make the thresholds adaptive to the gray-scale and lighting conditions of each processed image. A separate line of works [18, 19] went in a different direction and investigated the idea of directly detecting object boundaries. Lastly, graph-based instance image segmentation was proposed by Felzenszwalb et al. [20], where the similarity between each pair of pixels was evaluated to find where to cut the edges and split the different instances.

With increased computational power and the availability of bigger datasets, learning-based methods gained popularity, especially in complex scenarios where manually tuned heuristics are difficult to design. Most of the classical machine learning techniques such as random forests [21, 22], K-means clustering [23, 24], support vector machines [25, 26], and also graphical models [27] were initially applied. Currently, the task is commonly addressed with neural networks [28, 29] based on convolutions (CNNs) or transformer architectures [30, 31]. Several architectural enhancements have been proposed, such as atrous convolutions [32] to extend the receptive field and the context considered for each pixel, or an image-pyramid strategy [33] to inspect the image at different scales.

Different neural networks have been directly tested in the agricultural domain. Champ et al. [34] investigate the capabilities of Mask R-CNN [2]. The approach consists of two steps: object detection and then the generation of a pixel-wise mask. Another well-known approach is PanopticDeepLab [1], which is a general-purpose architecture to perform panoptic segmentation, predicting a center for each object in the scene and offsets for every pixel that belongs to that object. Although offset predictions require post-processing to obtain the final instances, this method usually outperforms the embedding-based ones, i.e., models that predict a high-dimensional embedding for each pixel in the image and then clusters similar embeddings into single objects. Weyler et al. [10] compute instances using a similar methodology: they predict offset vectors for each pixel and cluster regions defined by covariance matrices predicted from the learned feature maps. Our previous work [8] targets a joint semantic, plant, and leaf instance segmentation

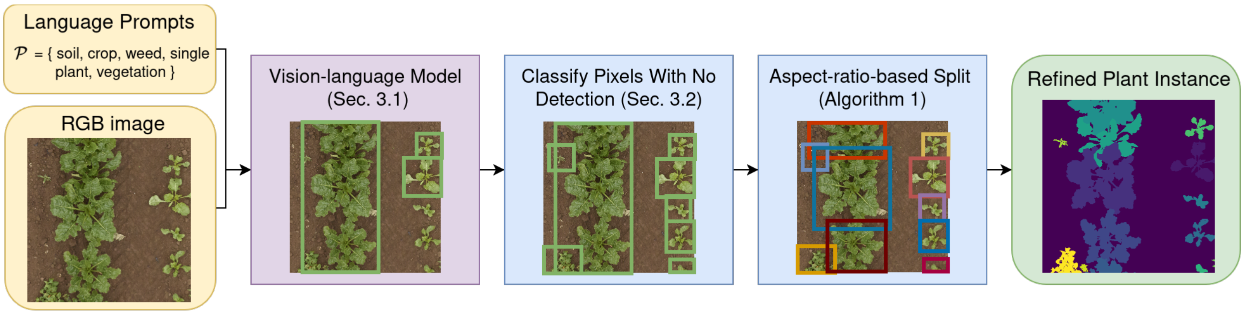

Language Prompts
𝓟 = { soil, crop, weed, single plant, vegetation }
RGB image
Vision-language Model (Sec. 3.1)
Classify Pixels With No Detection (Sec. 3.2)
Aspect-ratio-based Split (Algorithm 1)
Refined Plant Instance

Figure 2: Framework of our approach. A vision-language foundation model (purple) performs a first instance segmentation from language prompts and RGB images (yellow). Then, we use domain-specific heuristics (blue) to classify pixels without detections and to split instances of overlapping plants. The output (green) is a plant instance segmentation, where each color corresponds to a different instance.

using a similar methodology: predicting centers and offsets. The architecture leverages the natural hierarchy of these three tasks to improve the final performance.

However, both heuristics- and learning-based approaches often require to be adapted or re-trained to achieve a satisfactory performance on a new crop species or field. This depends on the diversity of the crop varieties, the common "closed world" assumption of the models that are trained on a small subset of classes, and on the ambiguous definition of crops and weeds in different agricultural settings, i.e., what is a crop in one field can be a weed in another.

In order to adapt such approaches, one usually needs access to vast amounts of labeled data. Several techniques have been developed in order to reduce the reliance on manual annotations by using pre-trained networks [35, 36, 37], i.e., initialized with weights optimized for a different task, weakly-supervised paradigms [38, 39], i.e., using partially or incompletely labeled data, or multi-modal foundation models [40].

Relying on general-purpose multi-modal foundation models trained on large datasets of paired texts and images is now a common way to address perception tasks. These so-called vision-language models (VLMs) [41] have competitive performance compared to fully supervised methods in many computer vision zero-shot tasks, i.e., tasks performed without adapting the model using additional training examples from the new domain. However, the performance deteriorates with increasing task complexity or when the application domain diverges too much from the original training dataset [42]. The domain gap is often tackled by fine-tuning the models on annotated data from the new application domain [43, 44].

In the context of agricultural applications, Shinoda et al. [45] introduced an evaluation benchmark for detection and classification of crops and plant diseases, showing that vision-language models achieve promising results but struggle with fine-grained tasks. Awais et al. [46] address this limitation by means of an expert-tuning approach to build a dataset to align the vision-language models with the agricultural domain. Most existing approaches, including VL-PAW [47] and E-CLIP [48] follow the trend of re-training or fine-tuning vision-language models on in-domain datasets to improve their performance. Chong et al. [49] take a different direction by avoiding the open-vocabulary inference and combining SAM [50] with BioCLIP [51] to obtain zero-shot semantic segmentation on agricultural images.

In this article, we focus on plant instance segmentation using images obtained in the agricultural domain, specifically images of crop fields [52], as shown in Fig. 1. The goal of the task is to assign a different ID to every observed plant. This task is central to image-processing pipelines for high-throughput phenotyping, where accurate per-plant information enables downstream analysis [10, 53]. An automatic pipeline for plant instance segmentation directly supports precision-agriculture practices, including yield estimation and targeted application of water or fertilizers, thus reducing the waste of resources [54]. Plant instance segmentation is particularly challenging for both heuristic and learning-based methods, due to overlapping foliage and the irregular and complex shape of leaves. Advancing robustness in this task is essential for autonomous field-monitoring systems and for future autonomous in-field intervention that would rely on such perception systems.

In contrast to the general trend, we propose to fuse the capabilities of existing vision-language models and background knowledge of the agricultural fields to label plant instances without using labels. This allows us to avoid re-training the vision-language model and to limit the hyperparameters of the heuristic-based post-processing that would need to be adapted to every new scenario. We predict a first instance segmentation using the vision-language model Grounded SAM2 [55] and then refine the predictions using domain-specific heuristic post-processing to improve our generated labels. At the same time, we do not need to provide additional manually annotated images, reducing the cost of labeling while outperforming state-of-the-art automatic labeling methods. Our pipeline can be a useful tool to automatically label images and use them to train or adapt fully supervised methods on the desired field setting.

We show in extensive experimental evaluation that our approach can be a valuable asset to generate plant instance segmentation labels and that we can use our labels to train fully supervised deep learning methods, requiring fewer manually acquired labels and boosting their final performance.

As we will see in the experiments, our approach (i) generates better plant instance labels than other state-of-the-art automatic labeling methods, improving the vegetation Intersection-over-Union (IoU) of 15.9 and Panoptic Quality (PQ) of 9.3 percentage points in average; (ii) boosts the performance of neural networks when used as additional input; (iii) reduces the need for labels when used as ground-truth annotation, producing comparable results using half of the manually annotated images; and (iv) helps the network generalize better on different fields without ground-truth annotations, improving zero-shot PQ performance of 21 percentage points on average.

2. Method

We propose an unsupervised framework for plant instance segmentation that leverages vision-language models combined with a domain-specific post-processing. The vision-language model provides an initial instance segmentation for an input RGB image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ of height H and width W, by means of zero-shot textual prompts $\mathcal{P}$. Building on these coarse predictions, we introduce a heuristics-based post-processing method that corrects for the common problems of the VLM predictions, such as missing detections or overlapping plants that are merged into a single instance. Our method leverages in-domain crop knowledge and geometrical cues to refine the coarse predictions without requiring any manually annotated data, keeping the pipeline unsupervised. Fig. 2 illustrates the overall pipeline, which we describe in the following sections.

2.1. Zero-Shot Instance Segmentation via Vision-Language Models

The first step of our approach is built on Grounded SAM2 [55]. Although we illustrate our approach using Grounded SAM2, one of the state-of-the-art VLMs, any architecture producing instances could serve as an initial step. Our goal is not to evaluate the best approach, but to demonstrate how our domain-specific post-processing improves the results obtained from any detection pipeline. For the extraction of initial candidates, they employ Grounding DINO [56], that given an input image $\mathbf{I}$ and text prompts $\mathcal{P}$ generates bounding boxes BB_i for each object o_i in $\mathbf{I}$ conditioned on $\mathcal{P}$. We use $\mathcal{P} = \{\text{soil, crop, weed, single plant, vegetation}\}$, and by using multiple synonyms for the vegetation class enables the model to capture more accurately the different vegetation components [57].

Grounding DINO is a transformer-based architecture that, from each input pair $(\mathbf{I}, \mathcal{P})$, extracts image $\mathbf{X}_I \in \mathbb{R}^{N_I \times d}$ and text features $\mathbf{X}_{\mathcal{P}} \in \mathbb{R}^{N_T \times d}$, where N_I is the number of image tokens, N_T the number of text tokens, and d corresponds to the feature dimension. These features are fused as $\mathbf{X} = \mathbf{X}_I \mathbf{X}_{\mathcal{P}}^{\top}$ and then passed to a decoder to obtain the detected objects $\mathcal{O}$. The approach uses two thresholds, one on the confidence of the bounding box prediction and one on the alignment with the text prompts to filter out uncertain detections and undesired objects. We kept the standard value of 0.3 for both thresholds.

The filtered bounding boxes BB_i from Grounding DINO are the input for SAM2 [58] to extract a pixel-wise mask $\mathbf{M}_i$ for each bounding box. The mask has one associated "semantic class" which is the text prompt $p_i \in \mathcal{P}$ with the highest confidence score. For further details, we refer to the original paper [55].

2.2. Plant Instance Segmentation

The second step of our approach uses the outputs of the zero-shot instance segmentation and refines them. There are two main short-comings of the outputs from the previous step: (i) Grounded SAM2 detects objects but allows some pixels in the image $\mathbf{I}$ not to be part of any detection; (ii) because of the difficulty of correctly separating overlapping plants, some detections need a refinement step to assign a unique ID to each plant. We show in Fig. 3 both of these

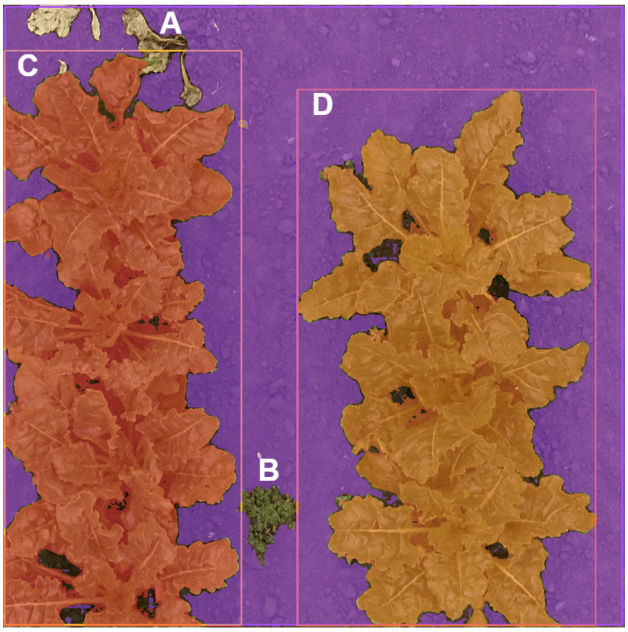


Figure 3: Output of Grounded SAM2, where soil is colored in purple and the vegetation instances are colored in different colors and surrounded by their bounding boxes. The leaves (A) in the upper-left corner is not segmented, as well as the weed (B) in between the two detected instances (C and D). Additionally, C and D both consist of multiple overlapping plants.

problems in an exemplary image from PhenoBench, where some leaves in the upper left corner (A) and a weed in the middle of the image (B) are not detected and where multiple plants are segmented as a single instance (C and D).

To solve the first problem, we compute the average RGB color for all pixels assigned to the vegetation and soil classes after the first step. We then use the cosine similarity to assign all not-segmented areas of $\mathbf{I}$ to the class – vegetation or soil – with the most similar color. Every area assigned to the vegetation class also gets a new instance ID. In this way, we correct for undefined objects in the field, i.e., stones, wires, and pipes, that we want to assign to the soil class, and for missing vegetation detection. At the end of this step, all pixels have a semantic class, and every vegetation pixel is part of an instance.

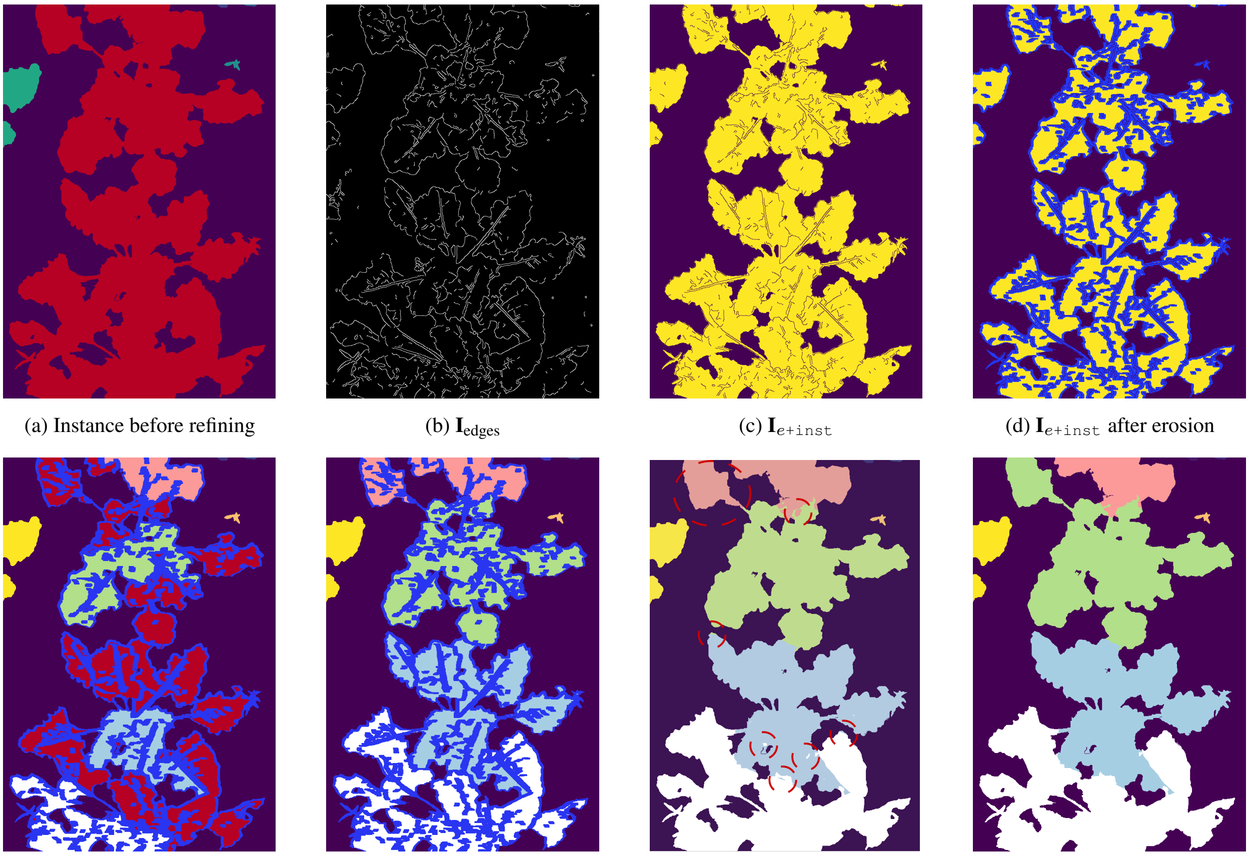
To solve the second problem, we need to detect which instances to refine. Using crop-specific knowledge, we can design a split function f that takes as input one instance binary mask $\mathbf{M}_i$ and return `True` if the instance needs to be split. In our implementation, we use Eq. (1) and the aspect ratio $a = \frac{H_i}{W_i}$ to detect if the instance needs to be refined. We define the split function as follows:

$$f(\mathbf{M}_i) = \begin{cases} \texttt{True} & \text{, if } a > \tau_a \\ \texttt{False} & \text{, otherwise} \end{cases} , \tag{1}$$

where τ_a is the aspect ratio threshold. At this point, we can also have an estimate of how many instances N_i have been aggregated according to our threshold, as

$$N_i = \lceil a/\tau_a \rceil . \tag{2}$$

We use the aspect ratio because it is independent of the size of the plant and the image resolution. In this way, we do not consider the number of pixels of the plant growth stage. Since the number of instances can only be an integer, in Eq. (2) we take the integer part of the result, which implicitly provides a tolerance, i.e., an aspect ratio $a = 1.3$ with a threshold $\tau_a = 1$ detects only one instance with a margin of 0.3 of difference in the ratio. We enforce the instance number to be at least one. If this assumption is violated and $N_i < 1$, it means that the crop row in our image is horizontal instead of vertical. To compute the real number of instances, we would need to rotate the image to have a vertical crop row which translates to compute the inverse of the aspect ratio $\frac{1}{a} = \frac{W_i}{H_i}$.



(a) Instance before refining

(b) $\mathbf{I}_{\text{edges}}$

(c) $\mathbf{I}_{e+\texttt{inst}}$

(d) $\mathbf{I}_{e+\texttt{inst}}$ after erosion

(e) Largest $N_{\texttt{inst}}$ components

(f) Assignment of all components

(g) Final points' assignment

(h) Mask after cleaning

Figure 4: Step-by-step images depicting how we address the splitting of instances. In (a), we see the unified instance in red, while in (b) we show its edges. (c) shows the results of the XOR operation between the instance mask and $\mathbf{I}_{\text{edges}}$. In (d), we can see the result of the erosion; eroded points are colored in blue. In (e), we show the largest components colored in white, azure, green, and pink, while in (f) we see the assignment of all the other components. After assigning the blue points using a voting mechanism, in (g), we use red dotted circles to highlight points assigned to one instance but not connected to it. In (h), we show the final instance segmentation after cleaning.

Algorithm 1 Post-processing for one instance mask

```
1:  Input: image $\mathbf{I}_{\text{RGB}}$, mask $\mathbf{M}_i$, aspect ratio threshold $\tau_a$, erosion kernel $\gamma$
2:  Output: instance image $\mathbf{I}_{\text{inst}}$
3:  $\mathbf{I}_{\text{inst}} \in \mathbb{N}^{H\times W} = \mathbf{0}$    ▷ initialize final instance image with zeros
4:  $a = \frac{H_i}{W_i}$, $N_i = \lceil a/\tau_a \rceil$    ▷ compute aspect ratio $a$ and number of expected instances $N_i$
5:  if $N_i < 2$ then    ▷ if we detect less than 2 instances, the oroginal mask is already the final instance
6:      assign all pixels in $\mathbf{M}_i$ to a new instance ID in $\mathbf{I}_{\text{inst}}$
7:      return $\mathbf{I}_{\text{inst}}$
8:  $\mathbf{I}_{\text{edges}} = \text{canny}(\mathbf{I}_{\text{RGB}} \odot \mathbf{M}_i)$    ▷ compute edges using Canny edge detector within the instance mask
9:  $\mathbf{I}_{\text{e+inst}} = \text{erode}(\mathbf{I}_{\text{edges}} \oplus \mathbf{M}_i, \gamma)$    ▷ exclude edges from instance mask and erode with kernel size $\gamma$
10: $\mathbf{C} = \text{conn\_comp}(\mathbf{I}_{\text{e+inst}})$, where $\mathbf{C} \in \{0,1\}^{H\times W\times N_c}$    ▷ extract binary masks of all $N_c$ connected components
11: $S = \text{TopN}_{\text{i}}(\mathbf{C}, N_i)$    ▷ take the $N_i$ largest components of $\mathbf{C}$
12: assign pixels of each component in $S$ to a new instance ID in $\mathbf{I}_{\text{inst}}$
13: assign pixels of all components not in $S$ to the closest instance ID based on distance between centers
14: for each pixel in $\mathbf{M}_i$ not assigned do    ▷ here we consider all the eroded pixels
15:     assign $p_k$ to the instance ID ocurring the most between its neighbors
16: for each instance ID in $\mathbf{I}_{\text{inst}}$ do    ▷ cleaning to enforce each instance is connected
17:     keep the largest connected component assigned to that instance ID
18:     assign all other components to the closest connected instance or a new instance ID
19: return $\mathbf{I}_{\text{inst}}$
```

Suppose multiple instances have been segmented together because they share a boundary. We need to detect the boundary to separate the instances. We refer to Fig. 4 to illustrate better the next steps, and to Algorithm 1 for the pseudo-code of the implementation. Fig. 4a shows one red instance that our approach decides to split. This means that after computing N_i at line 2 of our algorithm, $N_i \geq 2$. We can compute the edges $\mathbf{I}_{\text{edges}} \in \{0,1\}^{H\times W}$ from the original RGB image with any edge detector. In our implementation, we first apply a smoothing to the image and then use the edge detector by Canny et al. [59]. $\mathbf{I}_{\text{edges}}$ is shown in Fig. 4b and computed at line 6 of the algorithm. We then exclude the edges from the instance mask with a bit-wise XOR operation between $\mathbf{I}_{\text{edges}}$ and $\mathbf{M}_i$, and we call the output $\mathbf{I}_{\text{e+inst}}$ shown in Fig. 4c. Since we cannot guarantee that the edge detector finds smooth and optimal edges to separate our instances, we erode $\mathbf{I}_{\text{e+inst}}$ using a kernel of size γ. This will expand our edges and better separate the instance, as can be seen in Fig. 4d. These two operations are combined in line 7 of our algorithm, where the argument of the erosion is the result of the bit-wise XOR operation, denoted as $\oplus$. We apply connected components to $\mathbf{I}_{\text{e+inst}}$ and obtain a set of binary masks $\mathbf{C} \in \{0,1\}^{H\times W\times N_c}$, where N_c is the number of connected components detected, as shown in line 8 of the algorithm. Each binary mask in $\mathbf{C}$ corresponds to one connected component in $\mathbf{I}_{\text{e+inst}}$. At line 9, we select the N_i components with the largest areas as our starting point for the new instances. At line 10, we assign a new unique ID to the N_i detected instances, which we colored in white, light blue, green, and pink in Fig. 4e. We iteratively assign all the other components in $\mathbf{C}$ that have not been selected as starting instances to the closest new instance, computing the Euclidean distances between the centers of the components. The result of this iterative process is shown in Fig. 4f.

Now, we take care of pixels belonging to the original instance removed with the erosion, depicted in blue in Fig. 4f. At line 13, we compute for each of these pixels the set of neighbor pixels and which instance they belong to. Then, we assign the pixel to the instance that occurs the most in its set of neighbors, implementing a voting mechanism. The result is shown in Fig. 4g. As both the Euclidean distance assignment and the voting mechanism do not consider connectivity , we can see that there are pixels assigned to an instance but separated from it, which we highlight in red dotted circles. As the last step, we clean the instance mask. We keep the largest component for each instance ID as it is, while assigning the smaller components to an existing instance connected to it, or a new instance otherwise. This is performed at lines 14 – 16 in the algorithm, and the resulting refined instances are shown in Fig. 4h.

3. Results

3.1. Experimental Setup

The main focus of this work is a fully unsupervised pipeline for plant instance segmentation that exploits vision-language foundation models and domain-specific post-processing. The approach takes RGB images as input and computes plant instance annotations that we use to (i) boost the performance of networks on data for which we have labels and (ii) improve the generalization of a network on different fields.

In Sec. 3.2, we show the results of different vision-language models and how our domain-specific post-processing improves their results on different agricultural datasets; then in Sec. 3.3, we show how to use our generated labels to improve the generalization capabilities and reduce the requirement for manually annotated data of fully supervised learning methods.

Datasets. We test our approach on three RGB agricultural datasets. Two of them are recorded on fields of sugar beets: one was introduced by Weyler et al. [10] (denoted as SugarBeets in the following) and the other is the public benchmark dataset PhenoBench [60]. The third dataset is GrowliFlower [61], which is recorded on a field of cauliflowers. The three datasets have different image resolutions, lighting conditions, and growth stages; furthermore, only PhenoBench provides weed annotations. We use the official validation and test sets of PhenoBench and GrowliFlower. The SugarBeets dataset consists of 745 images for training, 272 for validation, and 278 for testing.

Metrics. We compute the intersection-over-union (IoU) [62] for the vegetation, or as mean over crops and weeds. We also compute the panoptic quality (PQ) [63] to evaluate the quality of the instance segmentation.

Details and Hyperparameters. Our approach has two hyperparameters: the kernel size γ and the aspect ratio threshold τ_a. We fix the aspect ratio threshold $\tau_a = 1$ and the kernel size $\gamma = 3$, which is the default value in most libraries [64, 65]. We train all networks using the configuration suggested in their original papers unless they give different parameters for the specific dataset.

Baselines. We compare against heuristic-based approaches similar to our domain-specific post-processing, and the results of the vision-language models without our post-processing. In particular, we try two different object detectors, Grounding DINO [56] and Florence2 [66], and two versions of the Segment Anything Model [58], SAM2 and SAM2.1 [67]. These changes do not alter the input that we provide or the outputs that the foundation models provide to our domain-specific post-processing. Detailed information about the different object detectors and pipelines can be found in the original papers.

We use a general-purpose graph segmentation method for RGB images [20] as a first heuristic baseline, where each pixel is a node in the graph and all neighboring pixels share an edge. The approach uses multiple thresholds to decide where to split the graph according to a dissimilarity function computed on the RGB values of each pixel. The second heuristic-based baseline is the vegetation mask based on the hue histograms [68], which is a commonly used option that does not suffer from the changes in lighting and weather conditions affecting the RGB values of the images. The third one is the excess green index [69], where we use a threshold on the excess green index computed over the RGB values of the image to get a mask based on the predominance of the green color in the vegetation. For all the heuristic baselines, we compute the plant instance segmentation from the vegetation masks via connected components.

We use four deep-learning baselines for the experiments that exploits the generated plant instance labels. The first is Mask R-CNN [2] (denoted as MR from now on), a common choice for object detection and instance segmentation. It is a two-stage approach based on region proposals that are then processed to produce pixel-wise masks and classes for all instances. The second learning-based baseline is PanopticDeepLab [1] (denoted from now on as PDL), for which we use MobileNetV2 [70] as the backbone. PDL predicts offsets and centers for each instance and needs a post-processing step to produce an instance mask. We compare against two domain-specific approaches for plant instance segmentation. Firstly, the approach by Weyler et al. [10] focuses on the post-processing stage, where they use covariances to specify areas close to the center of the plants where all pixels should be or point to. Our previous work HAPT [8] is used as a second baseline. It introduces a new hierarchical design for skip connections to exploit the features of three semantic tasks to boost the final performance.

3.2. Experiments on Unsupervised Label Generation

The first experiment evaluates the performance of our approach for label generation, i.e., the combination of foundation models and our domain-specific post-processing. The experiments show that our approach improves the

	Approach	SugarBeets		GrowliFlower		PhenoBench	
		IoU	PQ	IoU	PQ	IoU	PQ
heuristic	Felzenszwalb et al. [20]	58.1	47.8	62.7	35.2	68.3	3.9
	Hassanein et al. [68]	67.8	34.8	71.3	13.9	74.5	2.6
	Woebbecke et al. [69]	73.1	66.8	76.3	24.5	75.1	22.6
VLMs	Grounded SAM2 [55]	72.9	78.6	72.0	74.1	58.2	60.6
	Florence2 [66] + SAM2 [67]	33.4	47.5	78.3	61.3	59.6	44.2
	Grounded [56] SAM2.1 [67]	69.9	**86.3**	66.4	84.0	45.3	62.7
ours	Grounded SAM2 [55] + ours	**75.2**	78.1	88.1	79.0	77.3	66.3
	Florence2 [66] + SAM2 [67] + ours	72.2	75.4	80.9	82.9	62.9	**67.0**
	Grounded [56] SAM2.1 [67] + ours	75.1	83.3	**88.6**	**85.2**	**78.7**	66.0

Table 1: Results of the vegetation IoU (soil IoU is not considered) and PQ for all of the baselines and all the different datasets. In bold the best results for each metric and dataset. All results given in %.

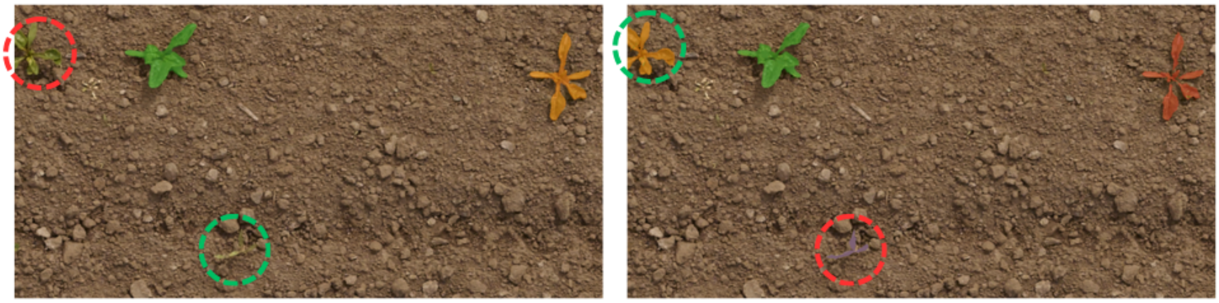


Figure 5: Qualitative images from Grounded SAM2 (left), and Grounded SAM2 + our post-processing (right). We highlight in red dotted circles the errors by the two approaches, and in green the correct prediction.

performance compared to only using the foundation models and outperforms the performance of heuristics-based methods. We note that we improve the IoU solving the first issue of VLMs, i.e. missing detections, and we improve the PQ using our heuristic-based post-processing to solve the second issue, i.e., overlapping plants.

Tab. 1 shows the results on all three datasets for all baselines. On the SugarBeets dataset, the VLMs and heuristics-based methods all have similar IoU results, except for Florence2, which produces fragmented masks. The other VLMs have a better PQ than all heuristic-based methods. Adding our post-processing improves the IoU in all investigated cases, but worsens the PQ for the models based on Grounding DINO, i.e., Grounded SAM2 and Grounded SAM2.1. We investigate this further looking at the qualitative results. In the image shown in Fig. 5 Grounded SAM2 had a vegetation IoU of 61.7% and a PQ of 95% since it only missed the plant in the red dotted circle. After our post-processing, the IoU is 75.5% because we are correctly identifying the missing plant as vegetation, but we also classify the weed at the bottom as vegetation. This error brings our PQ to 87.5%, since weeds are labeled as soil in the ground truth a correct vegetation detection for a weed is considered an error. This usually has a higher impact on the PQ than on the IoU because the number of pixels misclassified is low compared to the total number of vegetation pixels but the number of wrong detections is high compared to the number of total detections in the image.

For the GrowliFlower dataset, we show in Tab. 1 that all approaches yield good performance in differentiating vegetation and soil, probably thanks to good lighting conditions. However, the presence of various growth stages makes the instance segmentation task harder. GrowliFlower has some images with grass that should be detected as soil since it is not a crop to harvest or a weed to remove. It is hard for VLMs and heuristics-based methods alike to correctly classify grass as soil. We see that the VLMs have performance similar to the heuristic-based approaches regarding the IoU but superior results in the PQ. Using our domain-specific post-processing improves the results of all

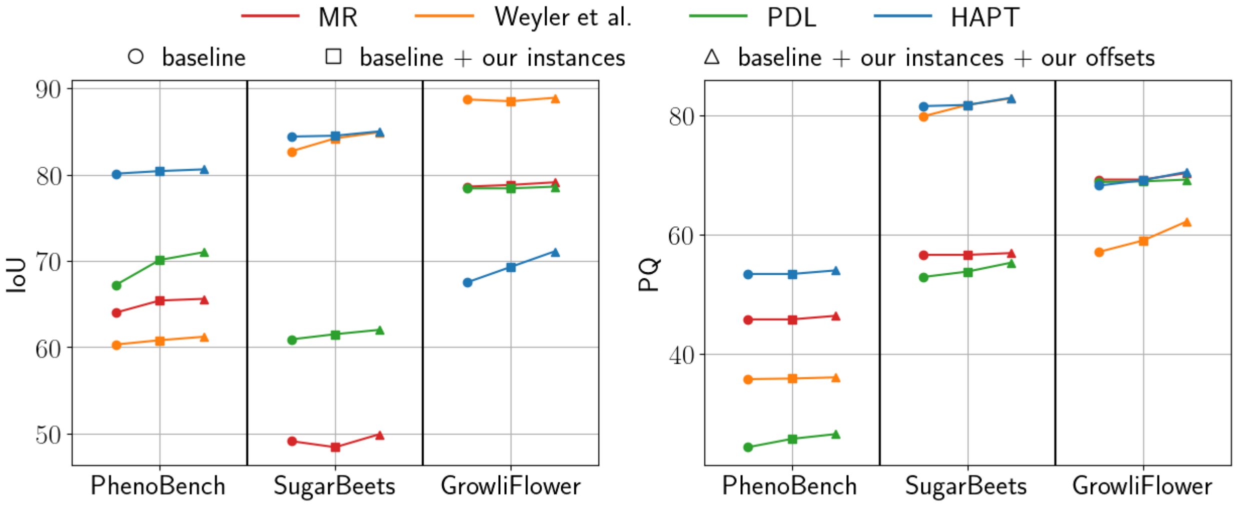


Figure 6: We show the results of the deep-learning networks (o) against the results obtained by the same networks when augmenting the input with our predicted instances (□), or with the predicted instances and the offsets computed from them in the x and y direction (△). Each colored line represents a different architecture to better show the trends of the metrics when using our additional inputs.

VLMs, both in IoU and PQ. The two most impressive results are the IoU of Grounded SAM2.1, improved by 22.2%, and the PQ of Florence2 + SAM2, improved by 21.6%. These improvements show that we correct both for missing vegetation detection and for wrongly merged instances.

The last columns of Tab. 1 show the results on the PhenoBench dataset, the only dataset that provides weed annotations. The dataset has images from late growth stages when many leaves overlap and create shadows, which makes the segmentation task challenging for both heuristic- and neural network-based approaches. The presence of shadows, weeds, and different growth stages limits the semantic segmentation of VLMs, all scoring lower IoU than heuristic-based methods. However, the VLMs have superior abilities in differentiating single plant instances, even with their reduced set of correct vegetation pixels. We see again that our approach improves the IoU and PQ of all methods, surpassing heuristic-based methods in terms of IoU and boosting the plant instance segmentation.

3.3. Experiments on Exploiting Our Generated Plant Instances Labels

In this section, we evaluate different ways to use our results to boost the performance of deep-learning approaches on the plant instance segmentation task. The results illustrate that: our approach (i) boosts the performance of neural networks when used as additional input; (ii) reduces the need for labels when used as ground-truth annotation; (iii) helps the network generalize better on different fields without additional ground-truth annotations.

3.3.1. Generated Instances as Additional Input

In this set of experiments, we can see that even when we have access to labeled data we can use our generated labels to improve the performance of learning-based systems. We conduct two types of experiments. First, we augment the input of the networks concatenating our labels to the RGB image as additional channel. Second, we ulteriorly concatenate the offset vectors for each instance we detected. To compute the offset vectors, we first determine the center of each instance. Then, for all pixels belonging to the instance, we calculate the difference of the pixel coordinates and the instance center. This vector represents the offset needed to move each pixel towards its instance center, enabling a clustering of the different objects. Since two of the segmentation baselines investigated, i.e., PDL and HAPT, directly predict instance offsets to perform instance segmentation, this information should be particularly beneficial for them. We evaluate this experiment on all three datasets and with all the learning-based approaches. We run all experiments under the same configuration and with a fixed random seed so that the only change is in the additional input provided.

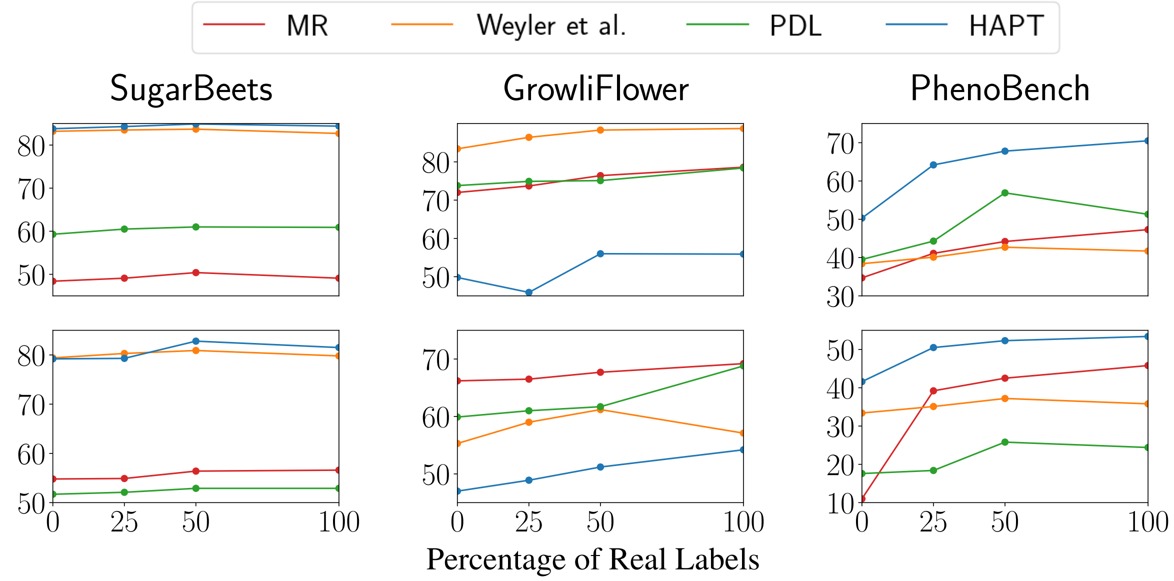


Figure 7: Results of three deep-learning networks, when using our approach instead of manual labels. On the x-axis we show the percentage of manual labels used during training. The first row depicts the IoU, i.e., crop on SugarBeets and GrowliFlower, and the mean of crop and weeds for PhenoBench. The second row shows the PQ.

Fig. 6 shows the results of the experiments on the three different datasets. We can see that over all experiments, the additional input boosts the final performance on both PQ and IoU. This is not true only in a few cases, and only for the IoU metric. MR with the predicted instances on SugarBeets and the approach by Weyler et al. [10] with the predicted instances on GrowliFlower have a lower IoU. However, the difference is in both cases below 1 %. Since the model with the best IoU can differ from the model with the best PQ, which we chose as our best model, picking the model with the best IoU would eliminate this problem at the price of having a smaller PQ improvement.

For the PhenoBench dataset, we can see that all approaches have improved their IoU and PQ when using our additional inputs. Since this dataset has crop and weed annotations, the IoU reported in Fig. 6 is the mean IoU over all the classes. It is interesting that for this specific dataset, it seems that the additional inputs are helping the IoU more than the PQ. We think this is because the PhenoBench dataset has high-resolution images $\left(1\frac{mm}{px}\right)$, and it presents small plants, which could be hard to detect for convolutional networks with big receptive fields. The additional inputs make small instances more visible to the network.

3.3.2. *Labels Substitution*

This experiment aims to show the capability of our approach to reduce the need for human-generated labels. For all datasets and all baselines, we run 3 different experiments, progressively substituting the humanly annotated training labels with the output of our pipeline. We run the experiments substituting 50%, 75%, and then 100% of the human-generated labels. We point out that the approach by Weyler et al. [10] is a bottom-up method that requires leaf instance labels as the first supervision. We decided to conduct the experiments with this approach by substituting only the plant instances with our approach and analyzing the results, knowing that the approach by Weyler et al. could still be capable of inferring the manual plant labels from the provided leaf instances.

In Fig. 7, we can see the results of the experiments. For SugarBeets and GrowliFlower, which only have crop and soil as semantic classes, we can see that even using less than half of the labeled data, the network can learn the task and perform well on the test set. The metrics are slightly lower than those obtained training on all the labeled data, but the difference of 2% points in performance is a good compromise if we need to label only 1/4 of the images. It is also interesting to notice that sometimes the metrics are better when using only part of the real labels. This effect is not consistent over the datasets or approaches: we can observe it in the IoU of MR for SugarBeets, in the IoU of HAPT and in the PQ of Weyler et al. for the GrowliFlower dataset, and in the metrics of PDL on PhenoBench. We visually investigated these results and concluded that considering the performance shown in Tab. 1, the error introduced by our labels is considered part of the data noise when there are enough manual labels to drive the learning-based approaches in the right direction. For the SugarBeets and GrowliFlower datasets, our generated labels are likely to have weed instances that can be considered hard negatives for the network to better learn the final task, leading to a small improvement.

3.3.3. Additional Labels

In the last set of experiments, we assess if scaling up the number of images in our training data by adding images annotated by our approach can help us to perform better. We evaluate this capability by testing the models on a joint test set made up of the validation set from PhenoBench and the test set from GrowliFlower. We train on the labeled data from PhenoBench and additional datasets labeled with our approach. We then evaluate how this diversity helps or degrades the performance on PhenoBench and GrowliFlower.

First, we use an additional sugar beets dataset introduced by Ahmadi et al. [71], which contains 287 images. This dataset provides a variation in the lighting conditions, growth stage, and image resolution, but not in the crop species. The ability to adapt to new scenarios, even if the objects in the scenes are the same, is part of what domain adaption algorithms try to solve. As a second experiment, we move more towards domain adaption, using a dataset of corn firstly presented by Ahmadi et al. [72] of 280 images. We point out that corn is not in our test set, containing PhenoBench and GrowliFlower, i.e., sugar beets, and cauliflowers. Nevertheless, we believe that a network can benefit from seeing different species of plants since our final goal would be to use it even on species that are not present in the training data. In the third experiment, we extend the training set with all the 1 542 images from the original GrowliFlower training set, using our predicted instance as instance labels. In this case, the data presented at training time has a more similar distribution to the test data, i.e., sugar beets and cauliflowers. The number of new images is comparable to the size of the original training set, so the networks should be able to optimize in a more fair way for both crop species. We run all experiments using MR, PDL, and HAPT and report the metrics in Tab. 2, Tab. 3, and Tab. 4. We cannot run the experiments on Weyler et al.'s approach since it needs supervision from the leaf instances and we cannot provide this for the additional data.

We can see that introducing additional data always improves the PQ of the combined evaluation set and the PQ and IoU of GrowliFlower. For MR and HAPT there are many experiments where the performance on the source domain gets worse, this is expected since the weights need to be optimized for new crop species, growth stages, and field conditions. PDL is the architecture that benefits the most from the additional data. This depends on the use of centers and offsets, which is more general than the region proposal of MR, and on the network's larger size making it less prone to overfitting. Looking for the best results, we can see that we obtain most of them using GrowliFlower or Corn as additional data; the first is expected since the training distribution matches the one for evaluation, while the second suggests that using different crop species can increment the ability of the networks to generalize even if the new species are not presents in the final evaluation data.

3.4. Ablation on Prompt Sensitiveness

In all of our experiments, we used the same set of prompts $\mathcal{P} = \{\text{soil, crop, weed, single plant, vegetation}\}$. As stated in Sec. 3, providing similar prompts with different phrasings ensures the model does not miss detections due to vocabulary mismatch and improves the chance that at least one prompt aligns with the visual features. In this section, we evaluate how different prompt sets affect the performance of the VLMs. We run all the experiments on the PhenoBench dataset and show the results of the VLMs without our post-processing in Tab. 5.

The results show that adding multiple prompts improves the panoptic quality. This trend suggests that the presence of multiple prompts helps the VLMs to correctly detect the single instances. To confirm this theory, we also tried to use $\mathcal{P} = \{\text{soil, crop}\}$ and $\mathcal{P} = \{\text{soil, single plant}\}$, to see if fewer prompts with high specificity were more suitable for our problem. The results were similar to the first lines in our table, with a mean of ± 2 percentage points of difference, confirming the need for multi-prompts to enable a more accurate plant instance segmentation. In the case of VLMs based on Grounding DINO [56], the increase of the PQ corresponds to a slight decrease in IoU, which is not visible for Florence2. This may suggest that the networks are optimizing for smaller instances, increasing the number of positive detections, but missing bigger instances that would contribute more to the IoU.

Extra Data	Test Set		IoU [%]			PQ [%]
	PB	GF	soil	crop	weeds	
none	✓		97.3	70.9	23.7	45.8
		✓	76.8	9.0	-	7.9
	✓	✓	90.5	50.2	23.7	32.8
Sugar Beets [71]	✓		(-0.5)	(-4.2)	(+16.2)	(+1.0)
		✓	(+1.2)	(+6.9)	-	(+2.9)
	✓	✓	(+0.0)	(-0.4)	(+16.2)	(+2.0)
Corn [72]	✓		(-0.4)	(-7.7)	(+16.3)	(-0.3)
		✓	(+9.9)	(+25.4)	-	(+24.0)
	✓	✓	(+3.0)	(+3.4)	(+16.3)	(+7.8)
GF (Train) [61]	✓		(-1.1)	(-3.4)	(+8.8)	(-11.9)
		✓	(+4.6)	(+23.8)	-	(+27.2)
	✓	✓	(+0.8)	(+5.8)	(+8.8)	(+1.2)

Table 2: Results on the validation sets of GrowliFlower (GF) and PhenoBench (PB), both independently and together, for MR trained on the training set from PhenoBench, with additional labels provided by our approach on different crops.

Extra Data	Test Set		IoU [%]			PQ [%]
	PB	GF	soil	crop	weeds	
none	✓		99.0	81.5	21.0	24.4
		✓	78.4	35.5	-	30.8
	✓	✓	92.1	66.2	21.0	26.5
Sugar Beets [71]	✓		(-1.6)	(-1.2)	(+1.6)	(+6.1)
		✓	(+11.4)	(+29.9)	-	(+9.5)
	✓	✓	(+2.8)	(+9.1)	(+1.6)	(+7.9)
Corn [72]	✓		(+0.1)	(+3.2)	(+8.1)	(+12.0)
		✓	(+15.6)	(+43.0)	-	(+15.0)
	✓	✓	(+5.3)	(+17.4)	(+8.1)	(+13.0)
GF (Train) [61]	✓		(+0.0)	(+2.6)	(+6.8)	(+9.3)
		✓	(+16.7)	(+51.0)	-	(+26.0)
	✓	✓	(+5.6)	(+18.7)	(+6.8)	(+15.5)

Table 3: Results on the validation sets of GrowliFlower (GF) and PhenoBench, (PB) both independently and together, for PDL trained on the training set from PhenoBench, with additional labels provided by our approach on different crops.

Extra Data	Test Set		IoU [%]			PQ [%]
	PB	GF	soil	crop	weeds	
none	✓		99.2	90.5	50.4	53.4
		✓	84.0	0.0	-	0.0
	✓	✓	94.1	60.3	50.4	35.6
Sugar Beets [71]	✓		(-0.5)	(-4.1)	(-14.0)	(-5.1)
		✓	(+2.9)	(+45.6)	-	(+28.6)
	✓	✓	(+0.5)	(+12.5)	(-14.0)	(+6.1)
Corn [72]	✓		(-0.8)	(-1.0)	(-11.6)	(-4.5)
		✓	(+0.0)	(+24.7)	-	(+21.8)
	✓	✓	(-0.5)	(+7.6)	(-11.6)	(+3.3)
GF (Train) [61]	✓		(-0.4)	(-1.5)	(-23.2)	(-8.2)
		✓	(+10.0)	(+78.5)	-	(+39.4)
	✓	✓	(+3.1)	(+20.2)	(-23.2)	(+7.8)

Table 4: Results on the validation sets of GrowliFlower (GF) and PhenoBench (PB), both independently and together, for HAPT trained on the training set from PhenoBench, with additional labels provided by our approach on different crops.

3.5. Analysis on Aspect Ratio Threshold τ_a

In this section, we perform an analysis to validate our chosen aspect ratio threshold $\tau_a = 1$. We compute the aspect ratio of all instances in the three datasets that provide plant instance labels, and show the results in Fig. 8.

We notice that the SugarBeets dataset has a peak of instances with a smaller aspect ratio. This can be explained by the relatively early stages of the plants in this dataset, which develop two leaves on opposite sides. In contrast, GrowliFlower has several plants with a larger aspect ratio because the images are relatively close to the plants. This means that many plants are only partially visible in the image, thus producing an irregularity in the expected aspect ratio. We fitted a Gaussian to each distribution, to show that our threshold $\tau_a = 1$ is at the peak of all three distributions, validating our choice. In particular the three Gaussians have $\mu = [0.98, 1.03, 0.93]$ and $\sigma = [0.41, 0.42, 0.35]$ for PhenoBench, GRowliFlower, and SugarBeets respectively.

3.6. Analysis on Erosion Kernel γ

In this section, we perform an analysis to validate our chosen erosion kernel $\gamma = 3$. We use the prediction coming from the same baseline, fixing the random seed, and computing per-image metrics to remove any stochasticity in the average per batch. Thus, we change the value of the erosion kernel γ from 1 to 9 and report the PQ to evaluate our choice and the parameter's range.

We show the result in Fig. 9. Firstly, we notice that the chosen value of $\gamma = 3$ is the one with the best PQ. Secondly, we see that ulteriorly increasing the value of γ degrades the performance, but the decrease is not constant. The difference in PQ decreases, as if it is reaching a plateau: the difference between $\gamma = 6$ and $\gamma = 7$ is 0.5 percentage points, for $\gamma = 7$ and $\gamma = 8$ is 0.4, and for $\gamma = 8$ and $\gamma = 9$ is 0.2. This trend suggests that with severe erosions, we are probably removing most of the vegetation pixels except for the core centers of the instances. In this way, the computational time increases as we have to iterate over all eroded pixels, but the performance only slightly changes due to the estimated instance centers.

The maximum difference in PQ, between $\gamma = 3$ and $\gamma = 9$, is of 6.8 percentage points. We compute the normalized sensitivity coefficient of our approach with respect to the parameter γ as

$$\text{Sensitivity} = \frac{\%\text{ change in output}}{\%\text{ change in parameter}} = \frac{6.8}{200} \approx 0.034. \tag{3}$$

This means that a change of 1% in γ produces a variation of 0.034% in PQ, demonstrating the robustness of the approach to possible changes of the parameter.

VLM	Prompts $\mathcal{P}$	Vegetation IoU [%]	PQ [%]
Grounded SAM2	{soil, vegetation}	60.7	52.7
	{soil, single plant, vegetation }	**61.2**	55.0
	{soil, crop, single plant, vegetation}	59.8	55.4
	{soil, weed, crop, single plant, vegetation}	58.2	**60.6**
Florence2 + SAM2	{soil, vegetation}	11.1	8.7
	{soil, single plant, vegetation }	18.7	14.8
	{soil, crop, single plant, vegetation}	57.8	29.8
	{soil, weed, crop, single plant, vegetation}	**59.6**	**44.2**
Grounded SAM2.1	{soil, vegetation}	**59.7**	57.3
	{soil, single plant, vegetation }	59.6	61.9
	{soil, crop, single plant, vegetation}	54.2	62.1
	{soil, weed, crop, single plant, vegetation}	45.3	**62.7**

Table 5: Results of the different VLMs while changing $\mathcal{P}$ to test their sensitivity to the inference prompts. We underline the newly added prompt for each line.

4. Discussion

The experiments presented suggest that our pipeline generates plant instance labels for training networks in the absence of manual labels. Our approach performs comparably to state-of-the-art fully supervised deep learning approaches without the requirement for labels. When labels are available, our suggestion is to exploit our method to boost the performance and increase the generalization capabilities of the model.

We show in Fig. 10 some qualitative images from the different datasets. On the top row, we show the masks produced by the VLMs. On the bottom row, we show the masks after our domain-specific post-processing. We point out that this visualization does not consider the possible double detections of VLMs, i.e., when the same pixels belong to different masks, nor missed soil detections, i.e., in the rightmost image, some pebbles and rocks were not assigned to any class. However, we can still notice how our post-processing improves the results, both detecting missing vegetation and obtaining a clearer separation of overlapping plants.

The quantity of available labels and the testing conditions motivate the best way to use our pipeline. In the case of unknown testing field conditions, the best way to proceed is to train on one human-annotated dataset and integrate different fields labeled with our approach to improve the generalization abilities of the network, see Sec. 3.3.3.

We showed the strengths of our approach, but there are also some failure cases due to our assumptions. Fig. 11 illustrates two of these situations. Since we rely on the output of the VLM, if the resulting masks are very inaccurate, it is almost impossible for us to improve their result. The main reasons why VLMs can fail are: (i) they are often trained on web images that rarely include complex vegetation scenes; (ii) for the same reason, their vocabulary may not be well aligned with agricultural terms; and (iii) lighting conditions, shadows, and occlusions can make it hard to extract a correct mask. Our approach tries to solve some of these issues using multiple prompts and our domain-specific post-processing, but severe failures are still hard to recover from.

The first example shows a soil mask which is also assigned to part of the plants and does not capture the whole soil. It would be hard for our approach to correctly segment the remaining pixels since the color of vegetation and soil are not reliable because of the wrong detections of the VLM. One possible solution would be to compute a soil-vegetation mask based on heuristics to check the overlap with the predicted soil mask from the VLM. This can help reject severe failure cases, using a vegetation mask as a semantic mask and connected components to initialize the instances. We could compute the standard deviation of the colors of the mask's pixels and detect a severe failure when it is above a certain threshold. However, this would need several experiments to define a threshold without wasting correct detections.

In the second case, the plant was not fully visible in the image, so the computed instance does not respect our desired aspect ratio. The aspect threshold τ_a is also a limiting factor of our approach. We performed all experiments with the same threshold $\tau_a = 1$, however, different crop species or data acquisition procedures may require adapting

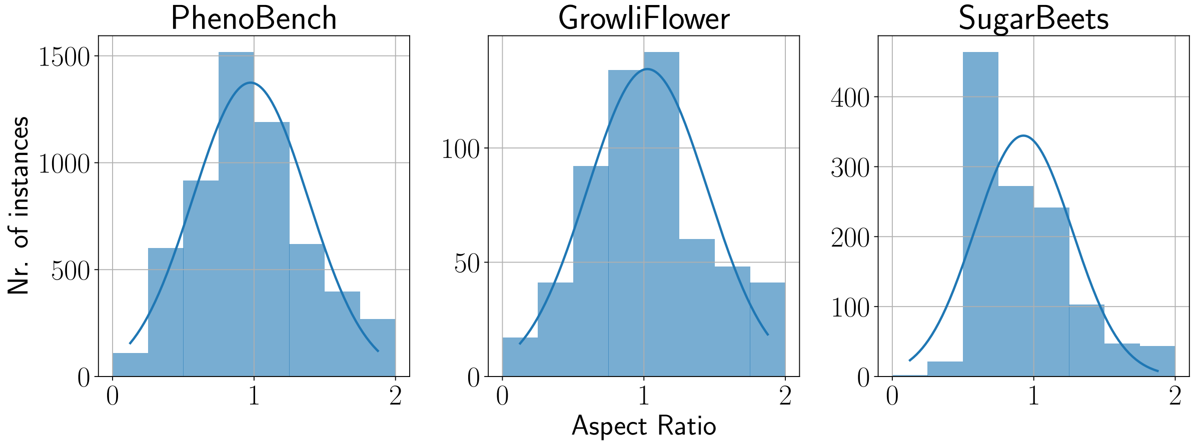


Figure 8: Histogram distributions of the aspect ratio of every plant instance in each dataset. We additionally plot the Gaussians estimated from the histograms.

the threshold to better capture the new expected shape of the crops. Finally, we know that our approach has difficulties finding weed instances when these are connected to plants because they are smaller compared to the plant size and can get eroded in our refinement step.

As our pipeline relies on the VLMs, it is complex to analyze the computational costs and inference speed. VLMs are usually large models between 500M and 2B of total parameters, making them computationally expensive to run on edge devices. The high number of parameters also makes their inference relatively slow, with times between 200 and 700 ms per image, depending also on the GPU and image size. This makes the pipeline unfeasible for real-time operations in the field as it is. However, recent works are focusing on reducing the inference time and computational complexity of VLMs [73], making them more accessible for real-world applications. Our post-processing, being heuristic-based, only counts two hyperparameters. Its runtime heavily depends on the VLMs' output, in particular on the areas without detections and the number of instances to split. For images of size 1024×1024 and initial instances coming from one of the three VLMs investigated in this article, our post-processing has a runtime of 31 ± 15ms on QUADRO RTX 5000 GPU.

5. Conclusion

In this article, we presented an effective approach to perform plant instance segmentation of RGB images. Our method exploits vision-language foundation models and domain-specific knowledge, improving the results of foundation models without requiring any additional annotated data. The benefit of our approach, however, goes further. It allows us to supervise learning-based approaches with unlabeled data as an initial training step boosting the performance of the deep learning systems without new manually provided labels. We implemented and evaluated our approach on different datasets and provided comparisons to other existing techniques, both heuristic- and neural network-based, and supported all claims made in this article. The experiments suggest that our approach is a competitive alternative to current state-of-the-art labeling methods for plant instance segmentation. Our method can generalize better than the baselines to unseen crop fields, making it applicable to new datasets without manually labeled training data. It can also be used as a pre-training step to initialize any instance segmentation network and achieve good performance when fine-tuning even on small human-annotated agricultural datasets.

Funding

This work has partially been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, EXC-2070 – 390732324 – PhenoRob

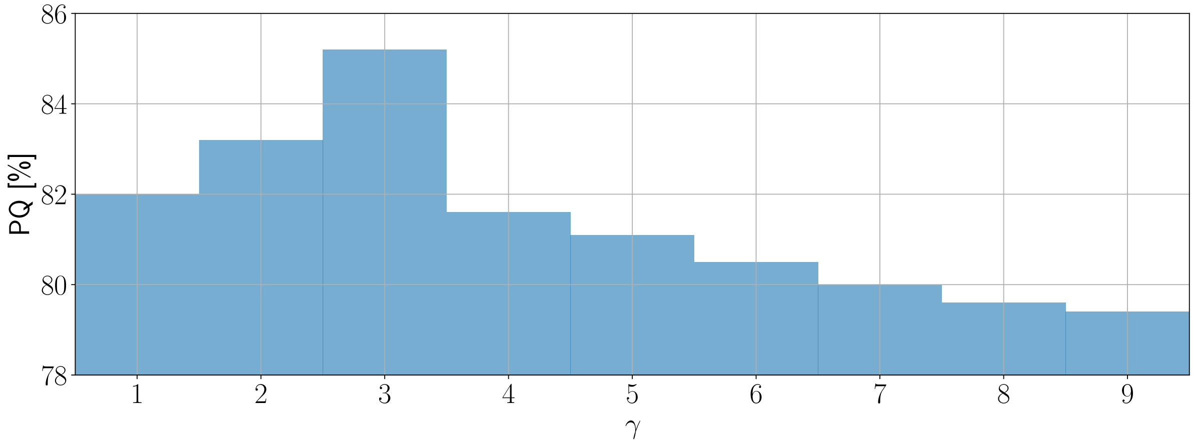


Figure 9: Results obtained varying the value of the erosion kernel γ when starting from the same VLM predictions.

Ethics Approval And Consent To Participate

Not Applicable.

Consent to Publish declaration

Not Applicable.

Data Availability Statement

The datasets are open and publicly available at https://rs.ipb.uni-bonn.de/data/growliflower/index.html for Kierdorf et al. [61], https://www.phenobench.org/ for Weyler et al. [60], https://uni-bonn.sciebo.de/s/HpUV7A1KofVop9u for Sugar Beets [71] and https://uni-bonn.sciebo.de/s/Eq0WVMa3y1uxB0h for Corn [71]. The original graph-based algorithm is available on https://cs.brown.edu/people/pfelzens/segment/; Mask R-CNN is at https://github.com/matterport/Mask_RCNN, PanopticDeepLab at https://github.com/facebookresearch/detectron2/tree/main/projects/Panoptic-DeepLab, HAPT at https://github.com/PRBonn/HAPT, and Weyler et al. at https://github.com/PRBonn/leaf-plant-instance-segmentation. We plan to publish the code and experimental setting for reproducibility of our experiments in the public GitHub repository of our institution https://www.github.com/PRBonn.

References

[1] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, L.-C. Chen, Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation, in: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2020.
[2] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV), 2017.
[3] R. Marcuzzi, L. Nunes, L. Wiesmann, J. Behley, C. Stachniss, Mask-Based Panoptic LiDAR Segmentation for Autonomous Driving, IEEE Robotics and Automation Letters (RA-L) 8 (2) (2023) 1141–1148.
[4] A. S. Chakravarthy, M. R. Ganesina, P. Hu, L. Leal-Taixé, S. Kong, D. Ramanan, A. Osep, Lidar panoptic segmentation in an open world, Intl. Journal of Computer Vision (IJCV) Special Issue on Open-World Visual Recognition (2024) 1–22.
[5] M. Sodano, F. Magistri, L. Nunes, J. Behley, C. Stachniss, Open-World Semantic Segmentation Including Class Similarity, in: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2024.
[6] M. Sodano, F. Magistri, T. Guadagnino, J. Behley, C. Stachniss, Robust Double-Encoder Network for RGB-D Panoptic Segmentation, in: Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA), 2023.

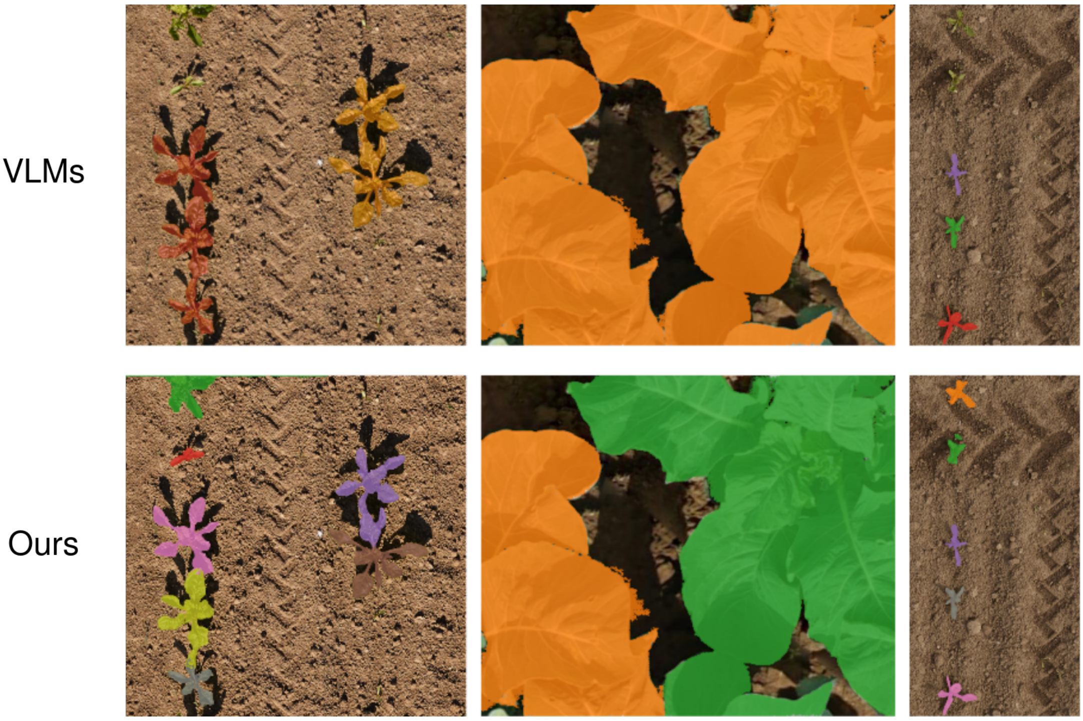


Figure 10: Qualitative results of the VLMs (top) and of the final masks obtained after our post-processing operation (bottom) for images coming from the different datasets. The results show improvements in IoU and in PQ, as our post-processing finds additional vegetation that was not detected and is able to separate plants that were assigned to the same instance.

[7] C. Yin, B. Wang, V. J. Gan, M. Wang, J. C. Cheng, Automated semantic segmentation of industrial point clouds using respointnet++, Automation in Construction 130 (2021) 103874.

[8] G. Roggiolani, M. Sodano, F. Magistri, T. Guadagnino, J. Behley, C. Stachniss, Hierarchical Approach for Joint Semantic, Plant Instance, and Leaf Instance Segmentation in the Agricultural Domain, in: Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA), 2023.

[9] R. Sheikh, A. Milioto, P. Lottes, C. Stachniss, M. Bennewitz, T. Schultz, Gradient and log-based active learning for semantic segmentation of crop and weed for agricultural robots, in: Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA), 2020.

[10] J. Weyler, F. Magistri, P. Seitz, J. Behley, C. Stachniss, In-field phenotyping based on crop leaf and plant instance segmentation, in: Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV), 2022.

[11] P. Lottes, C. Stachniss, Semi-supervised online visual crop and weed classification in precision farming exploiting plant arrangement, in: Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2017.

[12] G. Roggiolani, J.Rückin, M. Popović, J. Behley, C. Stachniss, Unsupervised Semantic Label Generation in Agricultural Fields, Frontiers in Robotics and AI 12 (2025).

[13] C. R. Brice, C. L. Fennema, Scene analysis using regions, Artificial Intelligence 1 (3-4) (1970) 205–226.

[14] F. Tomita, M. Yachida, S. Tsuji, Detection of homogeneous regions by structural analysis, in: Proc. of the Intl. Conf. on Artificial Intelligence (IJCAI), 1973.

[15] S. Wang, R. M. Haralick, Automatic multithreshold selection, Computer Vision, Graphics, and Image Processing 25 (1) (1984) 46–67.

[16] T. Pun, Entropic thresholding, a new approach, Computer Graphics and Image Processing 16 (3) (1981) 210–239.

[17] S. S. Reddi, S. F. Rudin, H. R. Keshavan, An optimal multiple threshold scheme for image segmentation, IEEE Trans. on Systems, Man, and Cybernetics 14 (4) (1984) 661–665.

[18] V. Caselles, R. Kimmel, G. Sapiro, Geodesic active contours, Intl. Journal of Computer Vision (IJCV) 22 (1997) 61–79.

[19] D. Chen, J.-M. Mirebeau, H. Shu, L. D. Cohen, A region-based randers geodesic approach for image segmentation, Intl. Journal of Computer Vision (IJCV) 132 (2) (2024) 349–391.

[20] P. F. Felzenszwalb, D. P. Huttenlocher, Efficient graph-based image segmentation, Intl. Journal of Computer Vision (IJCV) 59 (2) (2004) 167–181.

[21] F. Schroff, A. Criminisi, A. Zisserman, Object class segmentation using random forests, in: Proc. of British Machine Vision Conference (BMVC), 2008.

[22] C. Liu, R. Zhao, M. Pang, A fully automatic segmentation algorithm for ct lung images based on random forest, Medical Physics 47 (2) (2020) 518–529.

[23] N. Dhanachandra, K. Manglem, Y. J. Chanu, Image segmentation using k -means clustering algorithm and subtractive clustering algorithm, Procedia Computer Science 54 (2015) 764–771.

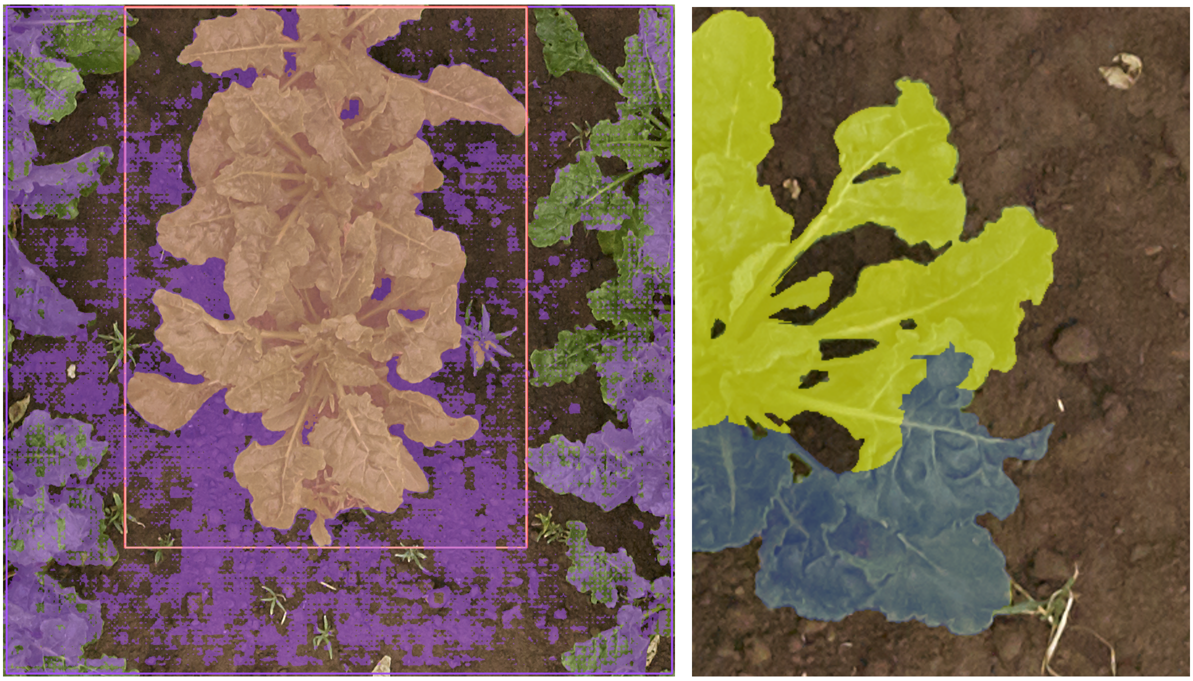


Figure 11: Two failure cases of our approach. On the left, the VLM fails in correctly detecting the soil, which is also assigned to most of the plants. Starting from this prediction makes it almost impossible to get a satisfactory plant instance segmentation. On the right, the instance is not respecting the expected aspect ratio because the plant is not fully visible, thus our approach splits it into two instances.

[24] S. A. Burney, H. Tariq, K-means cluster analysis for image segmentation, Intl. Journal of Computer Applications 96 (4) (2014).

[25] X.-Y. Wang, T. Wang, J. Bu, Color image segmentation using pixel wise support vector machine classification, Pattern Recognition 44 (4) (2011) 777–787.

[26] C. Kotropoulos, I. Pitas, Segmentation of ultrasonic images using support vector machines, Pattern Recognition Letters 24 (4) (2003) 715–727.

[27] L. Zhang, Q. Ji, Image segmentation with a unified graphical model, IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI) 32 (8) (2009) 1406–1425.

[28] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in: Proc. of the Conf. on Neural Information Processing Systems (NIPS), 2012.

[29] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016.

[30] X. Fu, Q. Ma, F. Yang, C. Zhang, X. Zhao, F. Chang, L. Han, Crop pest image recognition based on the improved vit method, Information Processing in Agriculture 11 (2) (2024) 249–259.

[31] X. Huang, D. Xu, Y. Chen, Q. Zhang, P. Feng, Y. Ma, Q. Dong, F. Yu, Econv-vit: A strongly generalized apple leaf disease classification model based on the fusion of convnext and transformer, Information Processing in Agriculture (2025).

[32] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. Yuille, DeepLab: Semantic Image Segmentation withDeep Convolutional Nets, Atrous Convolution,and Fully Connected CRFs, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 40 (4) (2018) 834–848.

[33] E. Adelson, C. Anderson, J. Bergen, P. Burt, J. Ogden, Pyramid methods in image processing, RCA Engineer 29 (1983) 33–41.

[34] J. Champ, A. Mora-Fallas, H. Goëau, E. Mata-Montero, P. Bonnet, A. Joly, Instance segmentation for the fine detection of crop and weed plants by precision agricultural robots, Applications in Plant Sciences 8 (7) (2020) e11373.

[35] P. Ge, C.-X. Ren, X.-L. Xu, H. Yan, Unsupervised domain adaptation via deep conditional adaptation network, Pattern Recognition 134 (2023) 109088.

[36] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum Contrast for Unsupervised Visual Representation Learning, in: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2020.

[37] A. Kolesnikov, X. Zhai, L. Beyer, Revisiting Self-Supervised Visual Representation Learning, in: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2019.

[38] J. Fan, Z. Zhang, Toward practical weakly supervised semantic segmentation via point-level supervision, Intl. Journal of Computer Vision (IJCV) 131 (12) (2023) 3252–3271.

[39] J. Ahn, S. Cho, S. Kwak, Weakly supervised learning of instance segmentation with inter-pixel relations, in: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2019.

[40] X. Yang, H. Dai, Z. Wu, R. Bist, S. Subedi, J. Sun, G. Lu, C. Li, T. Liu, L. Chai, Sam for poultry science, arXiv preprint arXiv:2305.10254 (2023).

[41] C. Li, Z. Gan, Z. Yang, J. Yang, L. Li, L. Wang, J. Gao, et al., Multimodal foundation models: From specialists to general-purpose assistants, Foundations and Trends in Computer Graphics and Vision 16 (1-2) (2024) 1–214.

[42] V. Udandarao, A. Prabhu, A. Ghosh, Y. Sharma, P. Torr, A. Bibi, S. Albanie, M. Bethge, No" zero-shot" without exponential data: Pretraining concept frequency determines multimodal model performance, in: Proc. of the Conf. on Neural Information Processing Systems (NeurIPS), 2024.

[43] S. Shao, Y. Bai, Y. Wang, B. Liu, B. Liu, Collaborative consortium of foundation models for open-world few-shot learning, Vol. 38, 2024, pp. 4740–4747.

[44] G. Han, S.-N. Lim, Few-shot object detection with foundation models, in: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 28608–28618.

[45] R. Shinoda, N. Inoue, H. Kataoka, M. Onishi, Y. Ushiku, Agrobench: Vision-language model benchmark in agriculture, in: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2025, pp. 7634–7644.

[46] M. Awais, A. H. S. A. Alharthi, A. Kumar, H. Cholakkal, R. M. Anwer, Agrogpt: Efficient agricultural vision-language model with expert tuning, in: Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV), 2025, pp. 5687–5696.

[47] G.-H. Yu, L. H. Anh, D. T. Vu, J. Lee, Z. U. Rahman, H.-Z. Lee, J.-A. Jo, J.-Y. Kim, Vl-paw: A vision–language dataset for pear, apple and weed, Electronics 14 (10) (2025) 2087.

[48] Y. Zhang, Y. Shao, C. Tang, Z. Liu, Z. Li, R. Zhai, H. Peng, P. Song, E-clip: An enhanced clip-based visual language model for fruit detection and recognition, Agriculture 15 (11) (2025) 1173.

[49] Y. Chong, L. Nunes, F. Magistri, X. Zhong, J. Behley, C. Stachniss, Zero-Shot Semantic Segmentation for Robots in Agriculture, in: Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2025.

[50] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, R. Girshick, Segment anything, in: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2023.

[51] S. Stevens, J. Wu, M. J. Thompson, E. G. Campolongo, C. H. Song, D. E. Carlyn, L. Dong, W. M. Dahdul, C. Stewart, T. Berger-Wolf, W.-L. Chao, Y. Su, Bioclip: A vision foundation model for the tree of life, in: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2024.

[52] J. Agrawal, M. Y. Arafat, Transforming farming: A review of ai-powered uav technologies in precision agriculture, Drones 8 (11) (2024).

[53] X. Tian, J. Wei, Z. Gao, Q. Zhao, Yolov9-screp: A lightweight instance segmentation and counting model for dense rice panicle images, Information Processing in Agriculture (2025).

[54] M. Minervini, H. Scharr, S. Tsaftaris, Image analysis: The new bottleneck in plant phenotyping, IEEE Signal Processing Magazine 32 (2015) 126–131.

[55] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, L. Zhang, Grounded sam: Assembling open-world models for diverse visual tasks, arXiv preprint arXiv:2401.14159 (2024).

[56] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, L. Zhang, Grounding dino: Marrying dino with grounded pre-training for open-set object detection, Vol. 15105, 2025, pp. 38–55.

[57] A. Gupta, P. Vuillecard, A. Farkhondeh, J.-M. Odobez, Exploring the zero-shot capabilities of vision-language models for improving gaze following, in: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 615–624.

[58] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, C. Feichtenhofer, Sam 2: Segment anything in images and videos, arXiv preprint arXiv:2408.00714 (2024).

[59] J. Canny, A computational approach to edge detection, IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI) 8 (6) (1986) 679 – 698.

[60] J. Weyler, F. Magistri, E. Marks, Y. Chong, M. Sodano, G. Roggiolani, N. Chebrolu, C. Stachniss, J. Behley, PhenoBench: A Large Dataset and Benchmarks for Semantic Image Interpretation in the Agricultural Domain, IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI) (2024).

[61] J. Kierdorf, L. V. Junker-Frohn, M. Delaney, M. D. Olave, A. Burkart, H. Jaenicke, O. Muller, U. Rascher, R. Roscher, Growliflower: An image time-series dataset for growth analysis of cauliflower, Journal of Field Robotics (JFR) 40 (2) (2023) 173–192.

[62] M. Everingham, L. Van Gool, C. Williams, J. Winn, A. Zisserman, The Pascal Visual Object Classes (VOC) Challenge, Intl. Journal of Computer Vision (IJCV) 88 (2) (2010) 303–338.

[63] A. Kirillov, K. He, R. Girshick, C. Rother, P. Dollár, Panoptic Segmentation, in: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2019.

[64] G. Bradski, The OpenCV Library, Dr. Dobb's Journal of Software Tools 120 (2000) 122–125.

[65] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, scikit-image: image processing in python, PeerJ 2 (2014) e453.

[66] B. Xiao, H. Wu, W. Xu, X. Dai, H. Hu, Y. Lu, M. Zeng, C. Liu, L. Yuan, Florence-2: Advancing a unified representation for a variety of vision tasks, in: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 4818–4829.

[67] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, et al., Sam 2: Segment anything in images and videos, 2025.

[68] M. Hassanein, Z. Lari, N. El-Sheimy, A new vegetation segmentation approach for cropped fields based on threshold detection from hue histograms, Sensors 18 (4) (2018) 1253.

[69] D. M. Woebbecke, G. E. Meyer, K. Von Bargen, D. A. Mortensen, Color indices for weed identification under various soil, residue, and lighting conditions, Trans. of the American Society of Agricultural Engineers 38 (1) (1995) 259–269.

[70] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks, arXiv preprint abs/1801.04381v3 (2018).

[71] A. Ahmadi, M. Halstead, C. McCool, Virtual temporal samples for recurrent neural networks: Applied to semantic segmentation in agriculture, Pattern Recognition (2021) 574–588.

[72] A. Ahmadi, M. Halstead, C. McCool, Bonnbot-i: a precise weed management and crop monitoring platform, in: Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2022.

[73] J. Fu, Y. Yu, N. Li, Y. Zhang, Q. Chen, J. Xiong, J. Yin, Z. Xiang, Lite-sam is actually what you need for segment everything, 2024.