# Leveraging on foundation deep neural models for individual apple tree segmentation in dense orchards via prompt engineering in RGB images

Anonymous ECCV 2024 Submission

Paper ID #19

**Abstract.** We propose a strategy to prompt a vision foundation model in order to address, in a few-shot learning mode, the segmentation of individual trees in dense orchards from simple RGB images. The method produces similar segmentation results to those of a classical supervised segmentation method.

**Keywords:** Promptable vision foundational model · Prompt engineering · Instance segmentation · Tree detection · Trunk detection · Phenotyping

## 1 Introduction

In modern computer vision for plant science, the recent literature is dominated by the use of supervised or self-supervised deep learning methods. While powerful, a limitation of these approaches is the lack of generalisation and the risk of overfit on the data used for the training. This is specifically the case in plant imaging in outdoor conditions due to the variability in lighting, the diversity of plant shape and shape complexity which evolves with plant growth. A solution for these limitations has recently appeared with the introduction of foundation models trained on extremely large amount of data (1 billion typically). The foundation models demonstrate very good capabilities of generalization on any type of data and excellent results when they are fine-tuned or guided with few prompts, i.e. indications. This development opens a new era in deep learning methods where the bottleneck is no more the annotation of images but the automation of prompt generation for effective use of the foundation models. The basic interest is that the time for annotation is considerably reduced since foundation models only need few shot or prompting to compete with the state-of-the-art standard supervised techniques. We follow this trend of prompt engineering for a specific task of plant imaging, which have been only very recently tackled with standard deep learning approach.

We focus on the segmentation of individual apple tree in dense orchards. Due to the high density of such orchards, adjacent branches may be inter-wined which makes instance tree segmentation a very challenging task. Most of the current literature addresses this challenge by processing point clouds generated from LIDAR data or point clouds generated from sets of RGB images [18, 33].

Very recently, authors have tested the possibility to perform tree segmentation with a single RGB image [10] based on classical supervised deep learning. In this article, we propose to investigate this specific task with a prompt engineering approach as depicted in the visual abstract of Fig. 1.

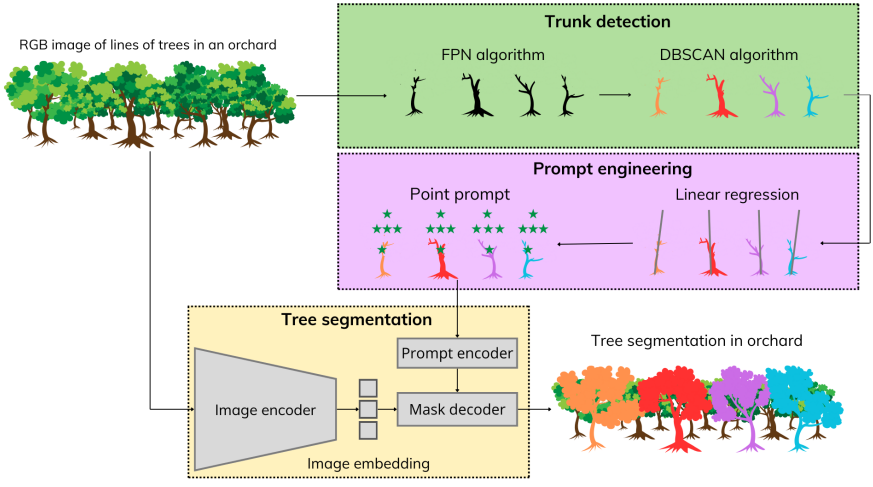


**Fig. 1:** Proposed workflow of individual tree detection algorithm. First a row of trees is photographed by an RGB camera (RGB image in orchards). In this image, an algorithm automatically detects tree trunks (Trunk detection). Then, points (prompts engineering) are defined in the space where the foliage of the tree is supposed to be, i.e. above each trunk. Finally, trees are automatically segmented from the prompts and the RGB image (Tree segmentation). The final result shows detected and segmented trees each marked with a different color.

## 2 Related work

Prompt engineering arises as a new paradigm with various strategies depending on the type of prompts and how they are used to boost few-shot learning [6, 22, 27, 29, 32]. Some specific prompt engineering strategies have been adapted to various application domains, as recently seen in medical imaging [1]. Similar approach is also deployed for plant imaging [2,4,17,24–26,30] and the proposition of this article lay in this trend. Related works corresponding to specific subparts of the proposed global pipeline of Fig. 1 are mentioned in the rest of the article.

# 3 Materials and Methods

The proposed workflow is structured into three parts: Trunk Detection, Prompt Engineering, and Tree segmentation. As input, we have a colour RGB image of a row of trees. As output, we obtain an instance segmentation mask. From an RGB image, a supervised deep neural model detects the tree trunks. This algorithm returns a binary segmentation mask (Figure 1-Trunk detection). Next, a regression line is drawn from each trunk, and points are defined above the trunks (Figure 1-Prompt engineering). These points will be used to guide the detection of a tree's foliage. Finally, a foundation model takes as input an RGB image and a set of points from the Prompt Engineering part (Figure 1-Tree detection). This model returns an instance segmentation mask. In this mask, each label identifies a tree. Figure 1 provides a visual abstract of our workflow.

## 3.1 Orchard description, data acquisition and annotation

The apple orchard is a REFPOP site [7] dedicated to variety testing (location will be provided after acceptance to avoid any conflict with the double blind review policy of CVPPA)

This site contains apple trees spread out over 14 rows on an area of 5057 $m^2$. Each tree is from a different variety. The apple trees are 7 years old. The length of each row is 90 meters. The inter-row distance is 4 meters. A row contains 86 trees organised in I-trellis structure with support poles. The distance between each tree is on average 1 meter. The height of the trees ranges from 1 to 3 meters. We assembled two data set from this REFPOP site. REFPOP 1 is composed of 7 full acquisitions of the same single row. REFPOP 2 is composed of single acquisitions of the rest of the remaining 13 rows. Furthermore, another data set from [10] is included in our study. Both REFPOP 1 and 2 were annotated manually including the shoot and the trunk. This data set serves as a reference data set for instance segmentation of apple tree in dense orchards from single RGB images. The images in the data set provided by [10] are centered on single trees. The trunk of neighbouring trees are not visible in the image. By contrast, the data set we propose with REFPOP 1 and 2 includes 25 times more trees than the data set of [10] and it includes 2 to 3 trees in each image. All images are standardised to have the same dimension 2000 × 1500 pixels. Table 1 provides the total number of trees in the images of the data set used for this study.

**Table 1:** Data set constructed in this study in comparison with the most related work.

| Data set | Number of images | Number of trees | Resolution |
|---|---|---|---|
| REFPOP 1 | 1462 | 3611 | 2000 × 1500 |
| REFPOP 2 | 444 | 861 | 2000 × 1500 |
| Most related work [10] | 150 | 150 | 2000 × 1500 |


**Fig. 2:** Sample of the data set of Table 1 used in this study. Left column stands for the input RGB images, middle column stands for the binary trunk mask ground truth, and right column stands for the ground truth for the segmentation output. First row is from REFPOP 1 tree row on a sunny day. The second row is from the REFPOP 2 on a cloudy day. The last row is from [10].

### 3.2 Trunk detection

The trunk detection method takes as input an RGB image of trees and returns a segmentation mask of the trunks. From these masks, it is possible to infer the direction of growth of the trees. This direction indicates the expected location of the foliage of the tree.

We considered several possible candidate methods from the literature to achieve trunk detection. In [12] a supervised method named TrunkNet, adapted for complex tree structures and the occlusion effects of trunks is proposed. However, this method seems to be trained with close up images of the trunk and the model is currently not made available yet. Alternatively [23] suggests a supervised segmentation method focused on separating tree branches while in our case the woody parts, apart from the trunk, are occluded by the foliage and are scarcely visible in the images. In a more related work [15] suggests a supervised segmentation method for the detection of the trunk and branches of apple trees. However, it requires colour and depth images while we use only RGB images.

Consequently, we have designed our own deep learning (DL) supervised segmentation method for the trunk detection. This method takes a colour image as input and returns a binary segmentation mask. It is structured in three steps: resizing the input image; determining trunk probability map over the image using a deep learning model; binarization of the mask by thresholding the probability map and applying the DBSCAN clustering algorithm [5, 21] to label individually each trunk. Any standard deep neural network for semantic segmentation could be used. We selected the VGG-16 as encoder and conducted tests with two encoder-decoder frameworks: UNet developped by [20] and LinkNet by [3]. Two models from two families of multi-scale models were also added: Feature Pyramid Network (FPN) developed by [13], and PSPNet by [31].

The detail of the data split are provided in Table 2. During training, we assess the prediction error of the model with the weighted Binary Cross-Entropy (wBCE) loss function and the quality of segmentation with the Dice coefficient. The wBCE loss function measures the discrepancy between the inferred probabilities of pixel belonging to a trunk and the ground truth, weighted by the normalized distribution of non-trunk pixels:

$$L_{BCE} = -\left(\sum_{i=1}^{H}\sum_{j=1}^{W} w_{not\ trunk} F_{(i,j)} \log p_{(i,j)} + w_{trunk}(1 - F_{(i,j)}) \log(1 - p_{(i,j)})\right)$$

where $(i, j)$ is the coordinate of a pixel, $H$ and $W$ are respectively the height and the width of images, $F$ denotes the function of ground truth labels and $p$ the inferred probability of a pixels belonging to a trunk. This loss function overcomes the problem of unbalanced trunk label distribution. The Adam algorithm was used to determine the optimal weights that minimize the loss function during training. The Dice coefficient similarity metric measures the area of the trunk pixels correctly predicted in comparison to the ground truth. Our neural network algorithm returns a probability map. A threshold $th$ was optimized on the validation data set. Thresholding the probability map returns a binary mask with the trunk label coded as 1 and the non-trunk label coded as 0. An unsupervised clustering algorithm is applied to the pixel locations classified as trunks in order to determine region of trunk instances. We chose to cluster pixels based on density using DBSCAN algorithm. Here again, we ajusted the hyperparameters (which allows for grouping common pixels together and discarding outlier pixels) on masks of the validation set. The trunk detection model was trained on PC equipped with an NVIDIA GTX 1660 SUPER GPU (21.9GB memory). The modeling was conducted using Python language code (v3.8.0) with TensorFlow library (v2.5.0). During training, the loss function was minimized using Adam algorithm with four parameters: learning rate equal to 0.0007, the decay rates pair $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 1 \cdot 10^{-7}$. The model was trained from scratch with a batch size of 4 and a total number of epochs equal to 100. The EarlyStopping function from Keras was used to stop the learning before overfitting in the last 5 epochs.

**Table 2:** Training-Validation-Test split composition. REFPOP 1* stands for a supbpart of REFPOP 1.

| Dataset | Split | Images |
|---|---|---|
| REFPOP 1 | Train | 128 |
| | Validation | 55 |
| REFPOP 1* | Test | 1269 |
| REFPOP 2 | | 444 |
| [10] | | 150 |

### 3.3 Prompt engineering

Once a tree is detected, we propose to detect the foliage via the use of a foundation model that need to be fed with prompts. We investigated the possibility to use points as prompts. We positioned these points in a rectangular area located just above the detected trunk. The width of the rectangle is set to be equal to the distance between two trees which can reasonably be considered as known for a given orchard. The height of the rectangle is set to be equal to a percentage of the expected height of the trees (we considered 80 %). Here again, this height can reasonably be considered as known for an orchard where all the trees have approximately the same age. Based on these priors we investigated 5 distinct strategies for the positioning of the points to serve as prompts. These strategies are illustrated in Fig. 4. We also tested various number of points in order to find a tradeoff: two few prompts might limit the quality of the segmentation while two much prompts increase the computational load to compute the model [16].

### 3.4 Tree segmentation

Concerning the promptable vision foundational model (VFM), we considered three possible candidates: CLIPseg developed by [14], SegGPT by [28] and SAM by [9]. Other models are corollaries of the three main ones mentioned above (see [8,11,19,34]) and could also be candidates. SAM is a segmentation method that requires points, bounding boxes or text as prompts to target the object to be segmented. As a semantic segmentation method, CLIPseg does not allow for the individual detection of trees. SegGPT only takes images and masks as input. We therefore picked SAM that can be used as a zero-shot object segmentation method with point prompt. Our method takes an RGB image as input and the points as input prompts. For a colour image, SAM will be applied to each tree. It returns the instance mask of the colour image. In this mask, each label corresponds to an individual tree.

### 3.5 Metrics

We evaluate three elements: the trunk segmentation, the number of correctly detected trunk and the segmented surface of the detected trunk. This was assessed

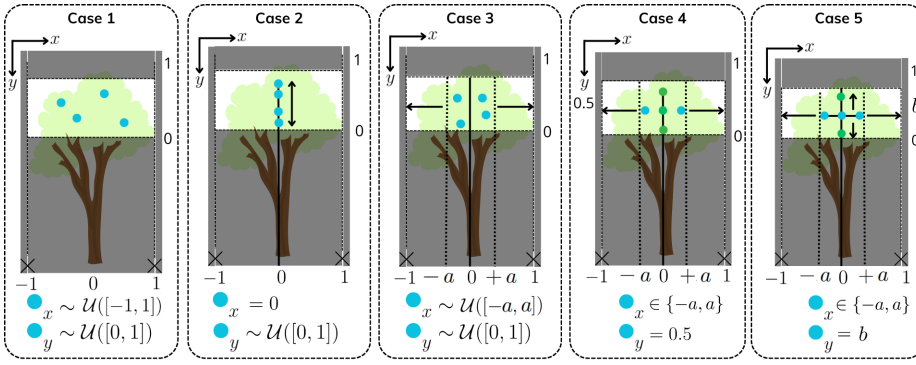


**Fig. 3:** Five approaches were tested to determine to determine an optimal prompt in the rectangle of interest. Case 1: Distribute points randomly within the rectangle of interest. Case 2: Distribute points in regular intervals along the regression line. Case 3: Distribute points randomly within an area that expands towards the edges (y-axis) of the rectangle with an hyperparameter a fixing the amplitude of the extension. Case 4: place three points at the top, middle, and bottom of the regression line. Two more points are added symmetrically on the line perpendicular to the regression line at an equal distance a. Case 5: same as case 4 but, here, the adjustable parameter is on the regression line and the two points placed perpendicularly at distance b.

with the following standard metrics. The trunk segmentation and the segmented surface of the detected trunk are measured with the Dice-Sorensen (d) metric together with the Precision and Recall. Precision is the number of well-predicted tree pixels among the labeled pixels. Recall measures how often the model correctly identifies positive instances (true positives) from all the actual positive samples in the dataset. The instance segmentation of all trees is assessed using the Average Precision (AP) metric. AP measures the average precision over the pixels assigned to a tree label. We also compute variants of AP with AP50 which is the average precision at Intersection over Union IoU = 0.5, AP75 the average precision at IoU = 0.75 and AP50-95 which stands for the average of the mean average precision calculated at varying IoU thresholds, ranging from 0.50 to 0.95. IoU measures the intersection between ground truth and segmented masks over the union of these two masks.

## 4 Results

We now present the results for the three parts of the method of Fig. 1 : trunk detection, prompts engineering, and tree segmentation.

### 4.1 Trunk detection

Table 4 shows the performance when the model is trained on REFPOP 1 and tested a variant of REFPOP 1* corresponding to an acquisition of the same

row but at a different time. This first performance serves as baseline to choose a segmentation network. Here the FPN (bold in Table 3) provides the best results. The following experiments will thus be conducted using this network. Also, the results of Table 3 demonstrates the difficulty of the task in terms of segmentation. Indeed REFPOP 1 and REFPOP 1* stands for the same row of trees acquired at two distinct time instances. As visible in Table 3, the performances are not perfect in terms of Dice. However, a high Dice is not our objective; rather we focus on the accuracy of trunk instances. Trunk segmentation only serves to the task of determining the regression line. Illustrations of the quality of the segmentation outputs is provided in Fig. 4.

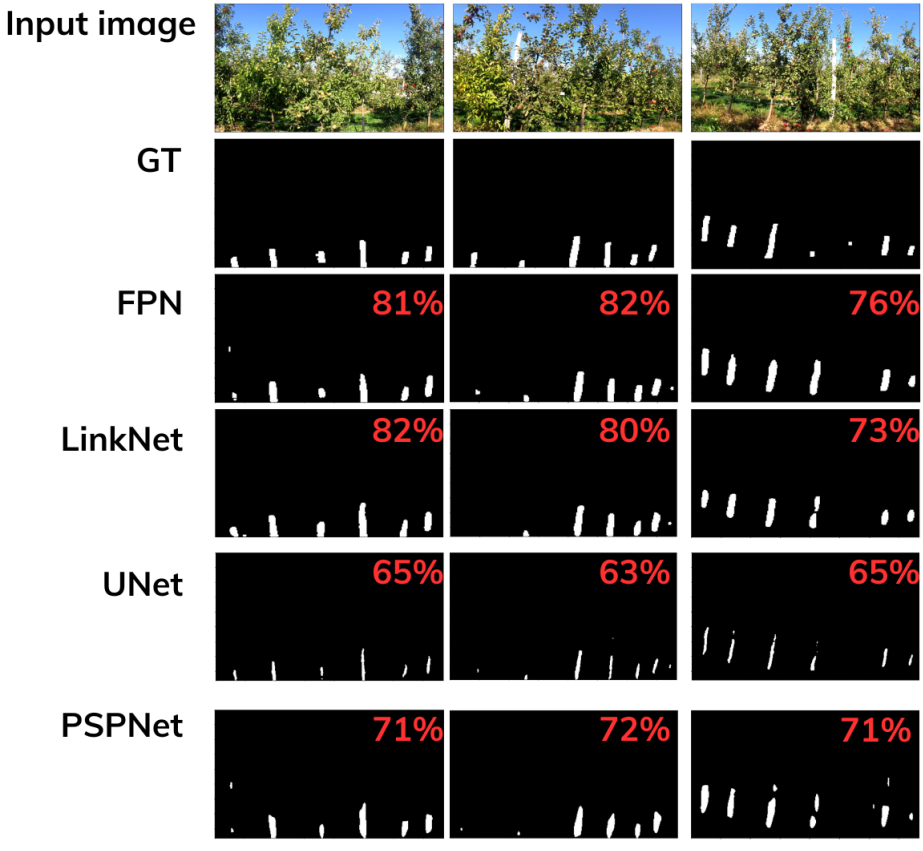


**Fig. 4:** Visual comparison for the four segmentation models. The red number at the top right is the recall.

To test the robustness of the segmentation trunk model of Table 3, we kept the best model and applied it to the rest of our test set. The results are provided

in Table 4. It appears that despite an important decrease of the Dice score the detection accuracy of the trunks is almost perfect.

**Table 3:** Trunk segmentation performance when tested on REFPOP 1*.

| | Dice | Precision | Recall |
|---|---|---|---|
| UNet | $0.72 \pm 0.09$ | 0.76 | 0.72 |
| LinkNet | $0.69 \pm 0.72$ | 0.72 | 0.71 |
| FPN | $\mathbf{0.73 \pm 0.11}$ | 0.75 | 0.75 |
| PSPNet | $0.63 \pm 0.11$ | 0.63 | 0.70 |

**Table 4:** Trunk segmentation of FPN model from Table 3 when tested on REFPOP 2 and the data set from [10].

| | Number of trunk predicted | Dice |
|---|---|---|
| REFPOP 2 | 861 over 861 | $0.52 \pm 0.18$ |
| [10] | 145 over 150 | $0.25 \pm 0.30$ |

### 4.2 Prompt engineering

Among the 5 approaches tested the case 5 demonstrated the best performance (exhaustive grid search details not shown) with following parameter values: width $a = 0.4$ (i.e. 40% of the average distance between trees) and height $b = 0.6$ (i.e. 60% the lowest point just above the detected trunk, the upper point at 80% and the middle point at 60% of the average height of the foliage). The optimal prompt configuration for our orchard is a diamond shape with an additional point in the middle of the diamond.

### 4.3 Tree segmentation

The average tree segmentation performance evaluation of our proposed segmentation pipeline are provided in Table 5 in comparison with the results obtained from the most related work in the state-of-the-art for this segmentation problem [10]. As visible in Table 5, our method stands below the performance of [10] except for $AP_{50:95}$ where we outperform [10] on our data set and perform the same on the data set of [10]. This is coherent with the initial observation by the author of [9] when comparing SAM with state-of-the-art supervised methods. It should be note that the state-of-the-art method [10] is a classical supervised method necessitating substantial number of training images while the pipeline proposed is a zero-shot learning of the SAM algorithm.

These results, while below the state-of-the-art in supervised learning, can be considered as encouraging when one analyzes the current segmentation errors. Examples of tree segmentation are provided in Fig. 5. Interestingly the segmentation of the forefront trees out of the background is not an issue. This demonstrates that LIDAR or 3D data are not mandatory to perform the segmentation of this forefront rows of trees even in the zero-shot learning mode we propose. The remaining errors appears for cases where trees are asymmetric. In these cases some of the lateral prompts may point to the wrong tree. In such cases, the average optimal prompt, which is symmetric is not adapted to fit with these singular shapes of each tree. To overcome this issue, one could use winter acquisition of the tree to optimize the prompting for each individual tree. This was done, for a different data modality, in [33]. Such an approach is applicable to RGB images from winter and summer; however it would require registration of images for each tree and thus comes with an additional computational cost.

**Table 5:** Performance of the pipeline of Fig. 1 when using SAM with the optimized prompt located just above the detected trunks.

| *SAM* | AP | $AP_{50}$ | $AP_{75}$ | $AP_{50:95}$ | Dice |
|---|---|---|---|---|---|
| REFPOP 1* | 0.93 | 0.90 | 0.82 | 0.90 | 0.57 ± 0.27 |
| REFPOP 2 | 0.90 | 0.91 | 0.90 | 0.91 | 0.70 ± 0.16 |
| Test Data from [10] | 0.83 | 0.83 | 0.82 | 0.83 | 0.84 ± 0.06 |
| Results from the method of [10] | – | 0.99 | 0.99 | 0.84 | – |

## 5 Conclusion and perspectives

We have proposed a prompt engineering pipeline leveraging on a foundation model to tackle the difficult problem of individual apple tree segmentation in dense orchards from sole RGB images. The performance, although below the state-of-the-art for supervised learning, is competitive considering the zero-shot nature of the segmentation tree step and demonstrates that LIDAR data may not be necessary to segment the forefront rows of trees from the background.

## References

1. Ali, H., Bulbul, M.F., Shah, Z.: Prompt engineering in medical image segmentation: An overview of the paradigm shift. In: 2023 IEEE International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings). pp. 1–4. IEEE (2023) 2
2. Carraro, A., Sozzi, M., Marinello, F.: The segment anything model (sam) for accelerating the smart farming revolution. Smart Agricultural Technology **6**, 100367 (2023) 2


**Fig. 5:** Example of tree segmentation based on our proposed method and comparison with ground truth: First - upper row from REFPOP 1, second - middle row from REFPOP2 and last - lower row from [10].

3. Chaurasia, A., Culurciello, E.: Linknet: Exploiting encoder representations for efficient semantic segmentation. In: 2017 IEEE Visual Communications and Image Processing (VCIP). IEEE (Dec 2017). https://doi.org/10.1109/vcip.2017.8305148, http://dx.doi.org/10.1109/VCIP.2017.8305148 5
4. Chen, Y., Yang, Z., Bian, W., Serikawa, S., Zhang, L.: Extraction study of leaf area and plant height of radish seedlings based on sam. In: Networking and Parallel/Distributed Computing Systems: Volume 18, pp. 69–83. Springer (2024) 2
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: kdd. vol. 96, pp. 226–231 (1996) 5
6. Gu, J., Han, Z., Chen, S., Beirami, A., He, B., Zhang, G., Liao, R., Qin, Y., Tresp, V., Torr, P.: A systematic survey of prompt engineering on vision-language foundation models. arXiv preprint arXiv:2307.12980 (2023) 2
7. Jung, M., Roth, M., Aranzana, M.J., Auwerkerken, A., Bink, M., Denancé, C., Dujak, C., Durel, C.E., Font i Forcada, C., Cantin, C.M., et al.: The apple refpop—a reference population for genomics-assisted breeding in apple. Horticulture research **7** (2020) 3
8. Ke, L., Ye, M., Danelljan, M., Liu, Y., Tai, Y.W., Tang, C.K., Yu, F.: Segment anything in high quality. In: NeurIPS (2023) 6

9. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. arXiv:2304.02643 (2023) 6, 9
10. La, Y.J., Seo, D., Kang, J., Kim, M., Yoo, T.W., Oh, I.S.: Deep learning-based segmentation of intertwined fruit trees for agricultural tasks. Agriculture **13**(11), 2097 (2023) 2, 3, 4, 6, 9, 10, 11
11. Li, F., Zhang, H., Sun, P., Zou, X., Liu, S., Yang, J., Li, C., Zhang, L., Gao, J.: Semantic-sam: Segment and recognize anything at any granularity. arXiv preprint arXiv:2307.04767 (2023) 6
12. Li, R., Sun, G., Wang, S., Tan, T., Xu, F.: Tree trunk detection in urban scenes using a multiscale attention-based deep learning method. Ecological Informatics **77**, 102215 (2023) 4
13. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017) 5
14. Lüddecke, T., Ecker, A.: Image segmentation using text and image prompts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7086–7096 (2022) 6
15. Majeed, Y., Zhang, J., Zhang, X., Fu, L., Karkee, M., Zhang, Q., Whiting, M.D.: Deep learning based segmentation for automated training of apple trees on trellis wires. Computers and Electronics in Agriculture **170**, 105277 (2020) 4
16. Mazurowski, M.A., Dong, H., Gu, H., Yang, J., Konz, N., Zhang, Y.: Segment anything model for medical image analysis: an experimental study. Medical Image Analysis **89**, 102918 (2023) 6
17. Osco, L.P., Wu, Q., de Lemos, E.L., Gonçalves, W.N., Ramos, A.P.M., Li, J., Junior, J.M.: The segment anything model (sam) for remote sensing applications: From zero to one shot. International Journal of Applied Earth Observation and Geoinformation **124**, 103540 (2023) 2
18. Ozdarici, A., Ozgun, A.: Using remote sensing to identify individual tree species in orchards: A review. Scientia Horticulturae **321**, 112333 (2023) 1
19. Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., Zeng, Z., Zhang, H., Li, F., Yang, J., Li, H., Jiang, Q., Zhang, L.: Grounded sam: Assembling open-world models for diverse visual tasks (2024) 6
20. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation (2015) 5
21. Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X.: Dbscan revisited, revisited: why and how you should (still) use dbscan. ACM Transactions on Database Systems (TODS) **42**(3), 1–21 (2017) 5
22. Shtedritski, A., Rupprecht, C., Vedaldi, A.: What does clip know about a red circle? visual prompt engineering for vlms. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11987–11997 (2023) 2
23. Silva, R., Junior, J.M., Almeida, L., Gonçalves, D., Zamboni, P., Fernandes, V., Silva, J., Matsubara, E., Batista, E., Ma, L., et al.: Line-based deep learning method for tree branch detection from digital images. International Journal of Applied Earth Observation and Geoinformation **110**, 102759 (2022) 4
24. Sun, J., Yan, S., Alexandridis, T., Yao, X., Zhou, H., Gao, B., Huang, J., Yang, J., Li, Y.: Enhancing crop mapping through automated sample generation based on segment anything model with medium-resolution satellite imagery. Remote Sensing **16**(9), 1505 (2024) 2

25. Swartz, L.G., Liu, S., Cozatl, D.M., Palaniappan, K.: Segmentation of arabidopsis thaliana using segment-anything. In: 2023 IEEE Applied Imagery Pattern Recognition Workshop (AIPR). pp. 1–5. IEEE (2023) 2
26. Torres-Lomas, E., Lado-Jimena, J., Garcia-Zamora, G., Diaz-Garcia, L.: Segment anything for comprehensive analysis of grapevine cluster architecture and berry properties. arXiv preprint arXiv:2403.12935 (2024) 2
27. Wang, J., Liu, Z., Zhao, L., Wu, Z., Ma, C., Yu, S., Dai, H., Yang, Q., Liu, Y., Zhang, S., et al.: Review of large vision models and visual prompt engineering. Meta-Radiology p. 100047 (2023) 2
28. Wang, X., Zhang, X., Cao, Y., Wang, W., Shen, C., Huang, T.: Seggpt: Segmenting everything in context. arXiv preprint arXiv:2304.03284 (2023) 6
29. Zhang, C., Puspitasari, F.D., Zheng, S., Li, C., Qiao, Y., Kang, T., Shan, X., Zhang, C., Qin, C., Rameau, F., et al.: A survey on segment anything model (sam): Vision foundation model meets prompt engineering. arXiv preprint arXiv:2306.06211 (2023) 2
30. Zhang, W., Dang, L.M., Nguyen, L.Q., Alam, N., Bui, N.D., Park, H.Y., Moon, H.: Adapting the segment anything model for plant recognition and automated phenotypic parameter measurement. Horticulturae **10**(4), 398 (2024) 2
31. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network (2017) 5
32. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. International Journal of Computer Vision **130**(9), 2337–2348 (2022) 2
33. Zine-El-Abidine, M., Dutagaci, H., Galopin, G., Rousseau, D.: Assigning apples to individual trees in dense orchards using 3d colour point clouds. Biosystems Engineering **209**, 30–52 (2021) 1, 10
34. Zou, X., Yang, J., Zhang, H., Li, F., Li, L., Wang, J., Wang, L., Gao, J., Lee, Y.J.: Segment everything everywhere all at once. Advances in Neural Information Processing Systems **36** (2024) 6