Is Segment Anything Model a revolution in precision agriculture?

Alberto Carraro[1] and Francesco Marinello[1]

TESAF Department, University of Padova

Abstract. Precision agriculture uses accurate identification and mapping of crop features by automated mechanisms. The use of computer vision techniques implemented by supervised deep learning systems to solve many precision agricultural problems necessitates large-scale data collection and prolonged ground truth annotation by humans. The so-called foundation models in artificial intelligence are therefore becoming more and more significant. Meta is working on a project called Segment Anything to provide a base model for image segmentation. Without extra training, it can accomplish zero-shot generalization to strange objects and images. This study evaluates the performance of Meta's Segment Anything Model (SAM) for the problem of semantic segmentation of objects in the context of precision agriculture.

Keywords: Precision agriculture · Meta's Segment Anything Model (SAM) · Image segmentation · Computer vision

1 Introduction

The main points that the reader should get from this paper are:

- There are large labelled datasets for computer vision tasks but they contain images of easily recognizable subjects (animals, cars, roads, etc.)
- Scientists can take open source architectures pre-trained on those large datasets but then they need to adapt them to their specific case, with additional ground-truth labelling (that requires experts) and additional training
- SAM requires no additional training and the authors claim it has learned the very concept of "object", disregarding any specific semantic context: how well can it perform on non-ordinary images?

Bisogna prendere spunto da

- https://www.sciencedirect.com/science/article/pii/S2214317323000112
- https://www.sciencedirect.com/science/article/pii/S0168169921002593/pdf

Precision agriculture is a broad field that aims to improve agricultural practices' efficiency, productivity, and sustainability through science and technology. One of the critical challenges in precision agriculture is accurately identifying and mapping crop features and conditions such as diseases, plant height, leaf area index,

and crop growth stage. Computer vision-based approaches have driven innovation in the agricultural industry and are nowadays used in scenarios such as pest identification [**?**], precision livestock farming [**?**], and many more. Machine vision technologies support farmers and industries, saving time and costs while reaching very good efficacy.

In computer vision, there are four primary tasks that can be combined and handled in different ways. One is image categorization, which arranges images according to a limited number of classes. Another is object detection, which creates bounding boxes, or tiny rectangles, around particular items in photos that fall into specified categories. Semantic segmentation seeks to categorize areas inside the images that have been labeled at the pixel level based on texture, color, and spatial distribution. Instance Segmentation, which combines object recognition and semantic segmentation, recognizes the various instances presented in the image together with their borders at the pixel level.

Since early applications of semantic segmentation, this task has been implemented in various ways. Depending on the lighting conditions and the sharpness of edges, various techniques have been adopted, such as colour space conversion and combination of colour channels [**?**]. Other implementations use machine-learning-based classification techniques such as decision trees [**?**] and clustering [**?**]. Deep Learning is a subset of Machine Learning, where the primary tools are Deep Neural Networks. The use of Deep Neural Networks for the solution of computer vision tasks reached enormous diffusion and success also due to the public availability of models like VGG [**?**], U-Net [**?**], SegNet [**?**], DeepLab [**?**]. These are Deep Neural Nets of various kinds that have been trained over large datasets. Table 1 summarizes the main deep learning models and the datasets they were trained on, together with the task they are meant to perform.

Open-source deep learning models have revolutionized the field of artificial intelligence by providing accessible and adaptable solutions for various tasks, allowing researchers and developers to utilize them for their own projects. A particularly successful principle in precision agriculture is that of Transfer Learning [**?**], offering significant advantages in developing robust and accurate models for various agricultural tasks. By leveraging pre-trained deep learning models on large-scale general image datasets, transfer learning enables the transfer of knowledge from the source domain to the target agricultural domain. This approach allows researchers to overcome challenges associated with limited annotated agricultural datasets and improve model performance by initializing the network with learned features. Several studies have demonstrated the effectiveness of transfer learning in precision agriculture tasks, such as crop disease detection [**?**], crop type mapping [**?**], and yield prediction [**?**]. Transfer learning not only accelerates model training but also enhances the generalization and adaptability of the models to different agricultural environments, ultimately contributing to improved crop management and increased agricultural productivity.

Transfer learning still requires some ground-truth labelling and additional training in order to specialize the chosen Deep Learning model for the domain at hand. For this last purpose there are dataset specialised on the agricultural domain like

Model	Dataset used for training	Computer Vision Task
AlexNet [**?**]	ImageNet [**?**] (14,197,122 images)	Image Classification
VGGNet [**?**]	ImageNet	Semantic Image Segmentation
ResNet [**?**]	ImageNet	Image Classification, Object Detection
U-Net [**?**]	Various medical imaging datasets: - MICCAI [**?**] [**?**], - ISIC [**?**]	Semantic Image Segmentation
DeepLab [**?**]	- PASCAL VOC [**?**], - Cityscapes [**?**] (5,000 images), - ADE20K [**?**] (20,210 images), - COCO [**?**] (118,000 images)	Semantic Image Segmentation
YOLO [**?**]	COCO	Real-Time Object Detection
SegNet [**?**]	- CamVid [**?**], - Cityscapes, - SUN RGB-D [**?**] (10,335 images), - ADE20K	Semantic Image Segmentation
EfficientNet [**?**]	ImageNet	Semantic Image Segmentation

Table 1: Most popular open source deep learning models and datasets used in computer vision tasks.

the PlantVillage [**?**] project containing 54,306 images of 14 crop species with 26 diseases (or healthy) made openly available.

There are also large-scale datasets commonly used for semantic segmentation in precision agriculture. Some of the largest and widely used datasets are:

1. Plant Phenotyping Datasets: These datasets focus on crop and plant analysis, providing pixel-level annotations for various plant structures.
   - PlantCLEF [**?**]: A collection of plant images with annotations for leaf, stem, and flower structures.
   - Plant Phenotyping Dataset [**?**]: Contains images of Arabidopsis plants with annotations for leaves and other plant components.
2. Crop Field Datasets: These datasets consist of aerial or satellite images of crop fields with annotations for different crops or objects of interest.
   - CropSeg [**?**]: A dataset with high-resolution aerial images of different crops, annotated at the pixel level.
   - Crop DeepLab Dataset [**?**]: Contains crop field images with fine-grained annotations for different crop types and objects.
3. Aerial Imagery Datasets: These datasets focus on aerial images captured by drones or satellites, providing annotations for various objects or classes of interest.
   - ISPRS 2D Semantic Labeling [**?**]: A large-scale dataset with high-resolution aerial images and annotations for building, road, and vegetation classes.
   - DeepGlobe Land Cover Classification [**?**]: Consists of satellite imagery with annotations for land cover classes, including agricultural areas.

These datasets offer substantial amounts of labeled data for semantic segmentation tasks in precision agriculture. Researchers often utilize them to develop and evaluate models for crop analysis, disease detection, weed detection, and other applications in precision agriculture.

Recently, segmentation foundation models have seen tremendous advancements in the field of natural image segmentation [**?**][**?**], enabling accurate and efficient segmentation of objects in a fully automatic or interactive way. These models are typically based on transformer architectures and leverage pre-trained weights to achieve state-of-the-art performance and unprecedented generalisation ability on a wide range of natural images. Among these the Segment Anything (SA) project [**?**] is a task, a model, and a dataset for image segmentation. The Segment Aanything Model (SAM) in particular has learned a general notion of what objects are and this understanding enables *zero-shot* generalization to unfamiliar objects and images without requiring additional training. SAM has been trained on the SA-1B dataset [**?**], the largest segmentation dataset to date, with over 1 billion masks on 11 million images.

SAM is designed and trained to be promptable, so its segmentation capabilities can be extended and transferred to new image distributions and tasks. We evaluate its capabilities on the specific tasks of segmenting objects inside the pictures of vines of the ESCA dataset [**?**].

2 Materials and methods

Our goal is to compare the performances of the foundational segmentation model SAM with no training to the previously available segmentation model U-Net, which requires manual annotation. The chosen task is semantic segmentation i.e. to classify the pixels of each image into two different semantic categories: foliage (foreground) and background. This task could be an important first part in a Machine Learing pipeline whose later stages focus on the analysis of the Region of Interest inside the pictures, which in this case we considered to be the area occupied by foliage. Figure 1 illustrates what we considered as foreground: all entities that were at the distance of within 1m from the camera.

2.1 Segment Anything

The Segment Anything Model (SAM) utilises a transformer-based architecture [**?**], which has been shown to be highly effective in natural language processing [**?**] and image recognition tasks [**?**]. Specifically, SAM uses a vision transformer-based **image encoder** to extract image features and **prompt encoders** to incorporate user interactions, followed by a **mask decoder** to generate segmentation results and confidence scores based on the image embedding, prompt embedding, and output token.

The prompt encoders are tailored for different user inputs. SAM supports four different prompts: points, boxes, texts, and masks. Each point is encoded by Fourier

Fig. 1: Foreground vs background.

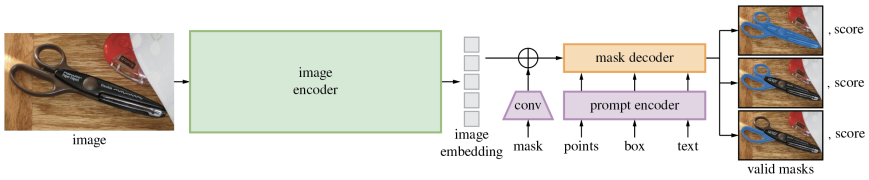
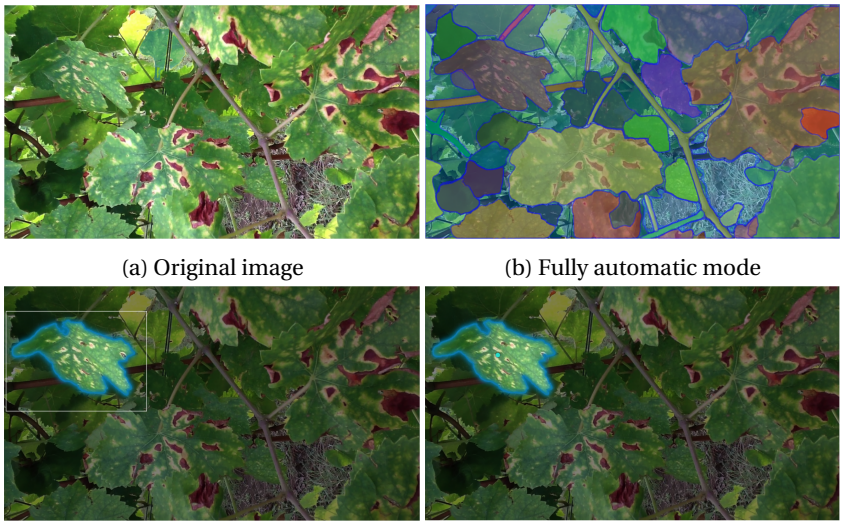


Fig. 2: Segment Anything Model (SAM) overview. A heavyweight image encoder outputs an image embedding that can then be efficiently queried by a variety of input prompts to produce object masks.

positional encoding [**?**] and two learnable tokens for specifying foreground and background, respectively. The bounding box is encoded by the point encoding of its top-left corner and bottom-right corner. The free-form text is encoded by the pre-trained text-encoder in CLIP [**?**]. The mask prompt has the same spatial resolution as the input image, which is encoded by convolution feature maps. Finally, the mask decoder employs a lightweight design, which consists of two transformer layers with a dynamic mask prediction head and an Intersection-over-Union (IoU) score regression head. The mask prediction head can generate three $4\times$ downscaled masks, which correspond to the whole object, part, and subpart of the object, respectively.

To summarize, when applying SAM for agricultural image segmentation, the segment-everything mode is prone to generate useless region partitions and the point-based mode is ambiguous and requires multiple prediction-correction iterations. In contrast, the bounding box-based mode can clearly specify the ROI and obtain reasonable segmentation results without multiple trials and errors. We argue that the bounding box-based segmentation mode has wider practical values



(a) Original image (b) Fully automatic mode

(c) Bounding box mode (d) Additional point modality

Fig. 3: Segmentation results of SAM based on different segmentation modes.

than the segment-everything and point-based mode when using SAM in medical image segmentation tasks.

Since the image encoder can be applied prior to prompting the model, we can pre-compute the image embedding for all training images to avoid replicated computing of the image embedding per prompt, which can significantly improve the training efficiency. The mask decoder only needs to generate one mask rather than three masks because the bounding box prompt can clearly specify the expected segmentation target in most situations.

2.2 U-Net and EfficientNet

U-Net is a Convolutional Neural Network (CNN) originally proposed for the segmentation of biomedical images [**?**]. Afterwards the same net has been used for tackling segmentation of Urban Environment surfaces from High Resolution Satellite Imagery [**?**]. Variations of U-Net have been used in precisoin agriculture, for example to segment cucumber leaves with disease spots [**?**], or images of Cichorium intybus L. root [**?**].

The U-Net architecture is, specifically, a Fully Convolutional Neural Network (FCNN) model commonly used for semantics segmentation and instance segmentation tasks. This architecture derives its name from its U-shaped structure (Figure 4) consisting of two main parts, from left to right: the contracting path (encoder) and the expanding path (decoder). The contracting path of U-Net consists of multiple convolutional and pooling layers. These layers gradually reduce the spatial dimensions while increasing the number of feature channels, which helps capture

a wide range of features at different scales. Each contracting block typically consists of two convolutional layers followed by a downsampling operation such as max pooling. The expanding path is the decoder part of U-Net and is responsible for generating the segmentation map. It consists of upconvolutional (also known as transposed convolutional) layers, which perform upsampling and increase the spatial dimensions. The expanding path also includes skip connections that concatenate feature maps from the corresponding contracting path. These skip connections enable the decoder to access the high-resolution feature maps from the encoder, aiding in the precise localization of objects. At the end of the U-Net architecture, a final convolutional layer is typically applied to produce the segmentation map with the desired number of output channels, representing the class probabilities or pixel-wise labels.

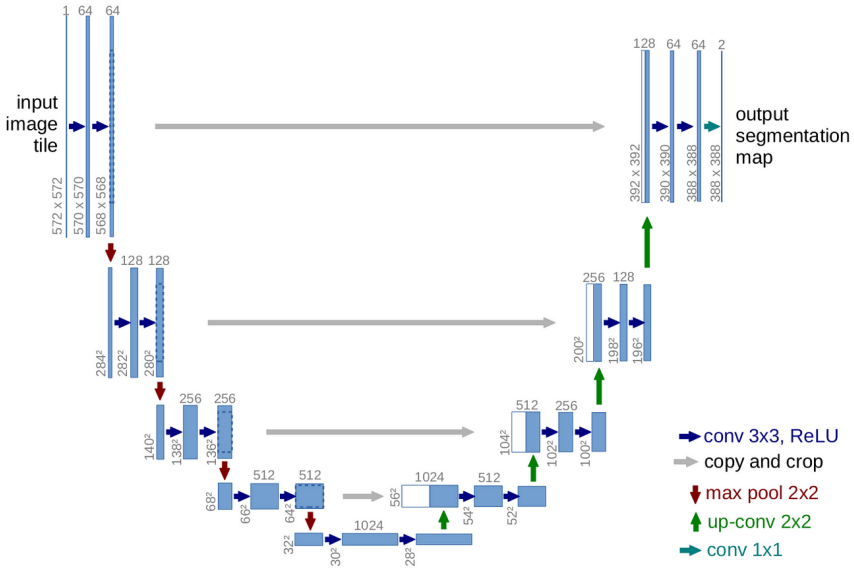


Fig. 4: U-net architecture

EfficientNet [**?**] is a family of convolutional neural network (CNN) models that are designed to achieve state-of-the-art performance while being highly efficient in terms of computational resources.

EfficientNet models are based on a concept called compound scaling, which optimizes the depth, width, and resolution of the network to achieve a good balance between accuracy and efficiency. The compound scaling technique involves uniformly scaling the network width, depth, and resolution using a coefficient called the compound scaling factor. This factor ensures that all dimensions of the network grow proportionally. By scaling up the network, it captures more complex patterns

and features, while scaling down reduces the number of parameters and computations required. EfficientNet models have achieved state-of-the-art performance on various computer vision tasks, including image classification, object detection, and segmentation, across different benchmark datasets. They have demonstrated superior accuracy compared to other popular CNN architectures, such as ResNet, while maintaining high efficiency in terms of memory usage and computational requirements.

EfficientNet models are often used as a backbone or encoder in various architectures, such as UNet, to improve performance and efficiency in tasks like image segmentation. The UNet architecture with an EfficientNet encoder combines the strengths of both models to achieve highly efficient and accurate image segmentation: it has been used in biomedical applications [**?**] [**?**] and it is implemented in the APEER ML toolkit by Zeiss® ([**?**]).

The UNet architecture, known for its U-shaped design, and EfficientNet, a state-of-the-art convolutional neural network (CNN) model, complement each other to improve performance. The EfficientNet encoder serves as the backbone of the network and is responsible for capturing high-level features from the input image. EfficientNet models are known for their superior performance and efficiency by leveraging compound scaling, which optimizes the depth, width, and resolution of the network based on a given resource constraint. In the UNet architecture, the contracting path (encoder) is responsible for capturing context and extracting features at different scales, while the expanding path (decoder) performs precise localization using skip connections. In the UNet with EfficientNet encoder, the contracting path is replaced with the EfficientNet backbone. The EfficientNet encoder provides a powerful feature extraction capability, capturing both low-level and high-level features from the input image. The extracted features are then passed to the expanding path, where upsampling and skip connections are used to recover the spatial resolution and refine the segmentation output.

3 Experiments and Results

The ESCA dataset [**?**] contains 1770 RGB images in jpg format taken at a working distance of approximately 30 cm from grapevine plants during sunny and windy days and considering scenarios with background variety. In 882 of these images, the totality of plants appearing therein is healthy (not affected by Esca). In contrast, the other 888 images contain at least one depiction of shoots with visible foliar symptoms of Esca disease. The ESCA dataset consists of two folders, esca and healthy, each containing images of the relative class.

Deep neural networks require large amounts of training data to tune millions of parameters and develop a learned model for subsequent predictions. While it is possible to crowdsource image annotation for natural scenes and collect large datsets as shown in Table 1, it is difficult to find experts to annotate images for specific scientific domains. Each research topic requires researchers to annotate their own images. Annotation being a very time consuming task, a researcher can end up with a handful of annotated images containing only tens of labeled objects.

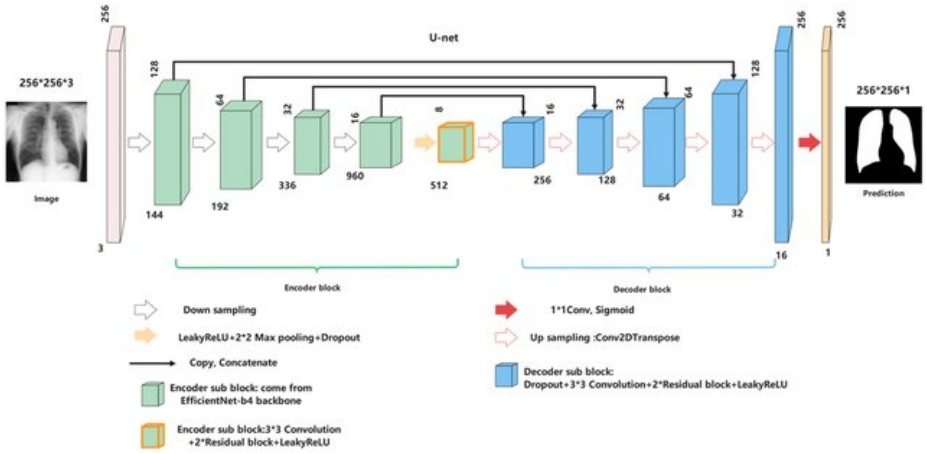


Fig. 5: U-Net architecture with EfficientNet encoder [**?**]

Our goal was to compare the performances of the foundational segmentation model SAM with no training to the previously available segmentation model U-Net, which requires manual annotation. The chosen task is semantic segmentation i.e. to classify the pixels of each image into two different semantic categories: foliage and background. This task could be an important first part in a Machine Learing pipeline whose later stages focus on the analysis of the Region of Interest inside the pictures, which in this case we considered to be the area occupied by foliage.

(a) (b)

Fig. 6: Ground truth segmentation of foliage (hand-made mask) for healthy1.jpg and esca1.jpg

We uploaded 102 images of the healthy folder and 103 of the esca folder to `https://www.apeer.com/annotate` and manually annotated them with the online semantic segmentation tool APEER Annotate [**?**], which allowed to classify the pixels of each image into two different semantic categories: foliage and background. The

reult of the manual sementation are a total of 205 binary masks in which the pixels corresponding to foliage area contain the value 1 and all others contain value 0.

Image name	Number of points on the border	Manual annotation time (seconds)
001_annotation.yaml	233	0
002_annotation.yaml	106	0
003_annotation.yaml	54	0
004_annotation.yaml	124	0
005_annotation.yaml	41	0
006_annotation.yaml	95	0
007_annotation.yaml	217	0
008_annotation.yaml	108	0
009_annotation.yaml	127	0
010_annotation.yaml	106	0
050_annotation.yaml	18	0
051_annotation.yaml	150	0
052_annotation.yaml	155	0
053_annotation.yaml	55	0
054_annotation.yaml	115	0
055_annotation.yaml	105	0
056_annotation.yaml	123	0
057_annotation.yaml	148	0
058_annotation.yaml	209	0
059_annotation.yaml	204	0
060_annotation.yaml	171	0
	Sum	**Sum**
	5904	0
	Average	**Average**
	98.4	0

Table 2: Time spent for drawing the masks manually (ground-truth labelling).

In Table **??** we recorded the time used to manually draw precise segmentation masks for 10 images. We keep the so-obtained 10 masks as a reference for the rest of the paper and consider them as ground-truth to measure performance of the two segmentation tools analyzed. In 3 we summarized the classification of pixel predictions in the context of semantic segmentation applied to the task of partitioning an image into its Region Of Interest (ROI) and its background.Frase contorta.

The metrics used in this paper are:

– Intersection over Union (IoU) or Jaccard Index: it measures the overlap between the predicted segmentation mask and the ground truth mask. It is calculated as the ratio of the intersection area to the union area of the two masks (in the case of two classes only this measure is equivalent to Mean Intersection over Union - mIoU).

Correct predictions	Incorrect predictions
True Positive (TP): the pixel has been predicted as part of the ROI and actually belongs to the ROI	False Positive (FP): the pixel has been predicted as part of the ROI but does not belong to the ROI
True Positive (TN): the pixel has been predicted as part of the background and actually belongs to the background	True Positive (FN): the pixel has been predicted as part of the background but actually belongs to the ROI

Table 3: Definitions of TP, TN, FP, and FN

- Dice Coefficient (DC): It measures the similarity between the predicted boundaries and the ground truth boundaries. It calculates the ratio of twice the intersection of the boundaries to the sum of their areas.
- Pixel Accuracy (PA): it measures the percentage of correctly classified pixels compared to the total number of pixels in the image. It calculates the overall accuracy of the segmentation.
- Precision, Recall, and F1-score: these metrics are commonly used in multi-class semantic segmentation evaluation. Precision measures the proportion of correctly classified positive pixels, recall measures the proportion of actual positive pixels that are correctly classified, and F1-score combines precision and recall to provide a balanced measure.
- Normalized Surface Distance (NSD): the average of the Hausdorff distance and the Symmetric Contour Distance. Hausdorff Distance: The Hausdorff distance measures the maximum distance between the predicted contour and the ground truth contour. It quantifies the largest discrepancy between the two contours and is sensitive to outliers or large deviations. Symmetric Contour Distance: The symmetric contour distance measures the average distance between the predicted contour and the ground truth contour. Unlike the Hausdorff distance, it takes into account both the under-segmentation (missing parts of the object) and over-segmentation (extraneous regions).

3.1 Segmentation of vine images with SAM

SAM supports three main segmentation modes: segmenting everything in a fully automatic way, bounding box mode, and point mode. The segment-everything mode divides the whole image into six regions based on the image intensity (Fig. 3b).

We segmented 10 images with bounding box mode and other 10 images with point mode. These modes are semi-automatic, i.e. they require a human input for each image.

Figure 3 shows the results of the three segmentation modes on an image of vines from the ESCA dataset [**?**] and Table **??** reports the exact time necessary to manually draw segmentation masks for the 10 selected images [1].

[1] The results are based on the online demo: `https://segment-anything.com/demo`.

Name	Formula
Intersection over Union (IoU)	
Dice Coefficient	
Pixel Accuracy	
Precision	
Recall	
F1-score	
Hausdorff Distance	$\max\left\{\max_{x\in\partial G} \mathrm{d}(x,\partial M), \max_{y\in\partial M} \mathrm{d}(y,\partial G)\right\}$
Average Symmetric Surface Distance	$\frac{\sum_{x\in\partial G} \mathrm{d}(x,\partial M) + \sum_{y\in\partial M} \mathrm{d}(y,\partial G)}{\lvert\partial G\rvert + \lvert\partial M\rvert}$

Table 4: Definitions of TP, TN, FP, and FN

We launched a batch segmentation job exploiting the fully-automatic mode on the entire set of 1770 images of the Esca dataset, using as hardware platform an NVIDIA RTX™ A6000 GPU. For the instantiation of the model we adopted the `ViT-H`[**?**] checkpoint. SAM produced for each segmented image a folder containing one mask for each object found in the image (between 100 and 200 objects). In order to process 888 images of the esca folder took 55 minutes and 13.8 seconds. In order to process 882 images of the healthy folder took 69 minutes and 19.4 seconds to complete the merge of masks took 48 minutes and 45.4 seconds to complete.

3.2 Segmentation of vine images with U-Net

In order to test semantic segmentation of vine images with U-Net we used the APEER ML toolkit by Zeiss® ?. The toolkit offers the possibility to train a U-Net with EfficientNet encoder that was then used to perform semantic segmentation on all 1770 images of the ESCA dataset. All generated masks were then downloaded.

4 Discussion and Conclusion

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla

Annotation file	# points inside mask	# points outside mask	Annotation time point mode (seconds)
001_annotation.json	41	20	101.696
002_annotation.json	35	21	95.565
003_annotation.json	32	22	106.857
004_annotation.json	46	29	125.596
005_annotation.json	43	49	148.657
006_annotation.json	53	71	210.004
007_annotation.json	72	63	201.255
008_annotation.json	58	68	186.949
009_annotation.json	59	81	193.657
010_annotation.json	51	68	197.857
050_annotation.json	19	138	98.891
051_annotation.json	118	176	388.167
052_annotation.json	98	171	276.081
053_annotation.json	32	140	114.474
054_annotation.json	67	120	254.87
055_annotation.json	75	138	337.862
056_annotation.json	60	180	338.139
057_annotation.json	77	167	293.152
058_annotation.json	95	184	310.982
059_annotation.json	79	166	338.914
060_annotation.json	94	187	276.054
	Sum	**Sum**	**Sum**
	3482	6268	12239.163
	Average	**Average**	**Average**
	58.033	104.467	203.986

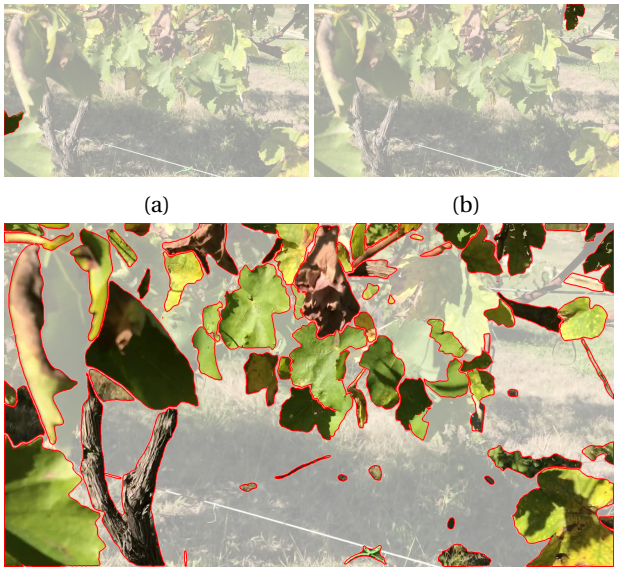
Table 5: Time spent to place points for SAM predictor.

		Image segmentation time (minutes)	
Image folder	**Number of images**	**Object segmentation (automatic mode)**	**Object masks merging**
Esca	888	55’ and 13.8”	???
Healthy	882	69” and 19.4”	48’ and 45.4”

Table 6: Image segmentation time with SAM in automatic batch mode.

pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

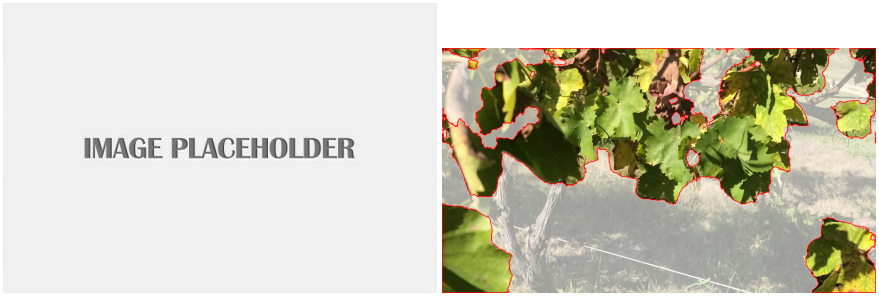
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla



(a) (b)

(c)

Fig. 7: Two masks produced by SAM in automatic mode on healthy1.jpg and the union of all masks produced for the same image.



(a) (b)

Fig. 8: Segmentation produced by U-Net on healthy.jpg and esca1.jpg

pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Image name	IoU	DC	PA	Prec	Rec	F1	ASSD (mm)	HD (mm)
001_image.png	0.804	0.892	0.962	0.157	0.887	0.266	6.456	328.868
002_image.png	0.779	0.876	0.971	0.101	0.89	0.182	5.887	262.574
003_image.png	0.829	0.907	0.985	0.072	0.872	0.134	6.274	329.474
004_image.png	0.804	0.892	0.976	0.098	0.857	0.175	2.832	147.872
005_image.png	0.861	0.926	0.991	0.054	0.931	0.102	3.275	237.306
006_image.png	0.769	0.87	0.976	0.081	0.806	0.147	2.999	127.44
007_image.png	0.819	0.9	0.968	0.145	0.896	0.25	2.595	170.88
008_image.png	0.81	0.895	0.979	0.089	0.923	0.162	3.451	164.657
009_image.png	0.815	0.898	0.981	0.081	0.933	0.15	5.787	361.334
010_image.png	0.792	0.884	0.983	0.064	0.939	0.12	5.45	311.045
050_image.png	0.276	0.433	0.966	0.013	0.805	0.025	150.533	588.205
051_image.png	0.846	0.916	0.975	0.138	0.912	0.24	6.189	499.401
052_image.png	0.803	0.89	0.974	0.106	0.928	0.19	11.023	485.956
053_image.png	0.722	0.839	0.978	0.056	0.944	0.106	32.785	456.054
054_image.png	0.89	0.942	0.988	0.099	0.919	0.178	2.697	469.998
055_image.png	0.802	0.89	0.979	0.085	0.884	0.155	11.072	358.236
056_image.png	0.789	0.882	0.972	0.103	0.882	0.184	7.121	238.481
057_image.png	0.81	0.895	0.977	0.096	0.905	0.174	6.389	218.009
058_image.png	0.867	0.929	0.98	0.132	0.9	0.231	4.626	173.807
059_image.png	0.756	0.861	0.967	0.101	0.88	0.181	7.385	252.634
060_image.png	0.858	0.924	0.983	0.102	0.924	0.184	5.307	319.265
	Avg.	**Avg.**	**Avg.**	**Avg.**	**Avg.**	**Avg.**	**Avg.**	**Avg.**
	.....	.....	.....	.....	.....	.....	.....	.....

Table 7: Evaluation of semantic segmentation performed by SAM in predictor mode with input points.

Acknowledgements The authors of this paper highly appreciate all the challenge organizers and owners for providing the public dataset to the community. We also thank Meta AI for making the source code of segment anything publicly available to the community.

Image name	IoU	DC	PA	Prec	Rec	F1	ASSD (mm)	HD (mm)
001_image.png	0.285	0.444	0.744	0.102	0.579	0.174	29.965	384.033
002_image.png	0.167	0.286	0.666	0.067	0.588	0.12	38.636	274.39
003_image.png	0.234	0.379	0.757	0.074	0.891	0.137	95.89	638.719
004_image.png	0.186	0.314	0.716	0.065	0.571	0.117	41.91	326.686
005_image.png	0.13	0.23	0.694	0.046	0.786	0.086	120.494	597.759
006_image.png	0.168	0.287	0.721	0.056	0.56	0.102	77.322	518.383
007_image.png	0.169	0.29	0.725	0.056	0.346	0.097	50.349	407.836
008_image.png	0.11	0.198	0.729	0.034	0.35	0.061	55.287	399.606
009_image.png	0.129	0.229	0.724	0.041	0.47	0.075	79.684	612.148
010_image.png	0.046	0.088	0.67	0.016	0.232	0.03	112.531	615.325
050_image.png	0.038	0.073	0.61	0.015	0.973	0.03	238.893	790.864
051_image.png	0.15	0.262	0.52	0.085	0.562	0.148	31.957	512.533
052_image.png	0.315	0.479	0.823	0.081	0.712	0.146	59.305	568.401
053_image.png	0.14	0.245	0.646	0.057	0.964	0.108	103.668	558.358
054_image.png	0.238	0.384	0.722	0.087	0.806	0.157	107.342	545.388
055_image.png	0.156	0.27	0.656	0.064	0.662	0.116	84.925	537.276
056_image.png	0.24	0.387	0.76	0.076	0.65	0.136	91.175	519.773
057_image.png	0.184	0.31	0.749	0.057	0.533	0.102	60.21	326.267
058_image.png	0.293	0.453	0.725	0.114	0.774	0.199	40.258	379.606
059_image.png	0.205	0.341	0.729	0.07	0.609	0.125	61.843	513.448
060_image.png	0.268	0.423	0.758	0.088	0.801	0.159	40.327	354.193
	Avg.	**Avg.**	**Avg.**	**Avg.**	**Avg.**	**Avg.**	**Avg.**	**Avg.**
	.....	.....	.....	.....	.....	.....	.....	.....

Table 8: Evaluation of semantic segmentation performed by SAM in fully automatic mode.

Image folder	Number of images	Image segmentation time
Esca	888	???
Healthy	882	???

Table 9: Image segmentation time with UNet

Image Name	IoU	DC	PA	Prec	Rec	F1
image1.jpg	...	...	...	...	...	...
image2.jpg	...	...	...	...	...	...
image3.jpg	...	...	...	...	...	...
image4.jpg	...	...	...	...	...	...
image5.jpg	...	...	...	...	...	...
image6.jpg	...	...	...	...	...	...
image7.jpg	...	...	...	...	...	...
image8.jpg	...	...	...	...	...	...
image9.jpg	...	...	...	...	...	...
image10.jpg	...	...	...	...	...	...

Table 10: Evaluation of semantic segmentation performed by UNet.