A Good Foundation is Worth Many Labels: Label-Efficient Panoptic Segmentation

Niclas Vödisch[1*], Kürsat Petek[1*], Markus Käppeler[1*], Abhinav Valada[1], and Wolfram Burgard[2]

***Abstract*—A key challenge for the widespread application of learning-based models for robotic perception is to significantly reduce the required amount of annotated training data while achieving accurate predictions. This is essential not only to decrease operating costs but also to speed up deployment time. In this work, we address this challenge for PAnoptic SegmenTation with fEw Labels (PASTEL) by exploiting the groundwork paved by visual foundation models. We leverage descriptive image features from such a model to train two lightweight network heads for semantic segmentation and object boundary detection, using very few annotated training samples. We then merge their predictions via a novel fusion module that yields panoptic maps based on normalized cut. To further enhance the performance, we utilize self-training on unlabeled images selected by a feature-driven similarity scheme. We underline the relevance of our approach by employing PASTEL to important robot perception use cases from autonomous driving and agricultural robotics. In extensive experiments, we demonstrate that PASTEL significantly outperforms previous methods for label-efficient segmentation even when using fewer annotations. The code of our work is publicly available at http://pastel.cs.uni-freiburg.de.**

***Index Terms*—Semantic Scene Understanding; Deep Learning Methods; Computer Vision for Transportation**

I. Introduction

HOLISTIC scene understanding is a core requirement for mobile robots to interact autonomously with their environment. Commonly, this is addressed by visual panoptic segmentation that assigns a semantic class to each pixel while separating instances of the same class. Although recent methods [1], [2], [3] have shown great progress in terms of segmentation performance, they often rely on a vast amount of densely annotated training data and tend to generalize poorly to new domains. Since creating panoptic labels is a highly laborious task [4], collecting large-scale training data for every new area of operation would drastically increase the cost of robot deployment. This particularly hinders the widespread application in continuously changing environments, e.g., agricultural robotics. To reduce training costs, some recent segmentation techniques employ various kinds of limited supervision. For instance, by learning from sparse annotations [5], [6], in semi- [7], [8], [9] or unsupervised manners [10], [11], and more recently by leveraging foundation models [12], [13]. Since these models can be adapted to various downstream tasks [14], [15], we argue that they offer a powerful pretraining strategy for addressing robotic perception tasks in a label-efficient manner.

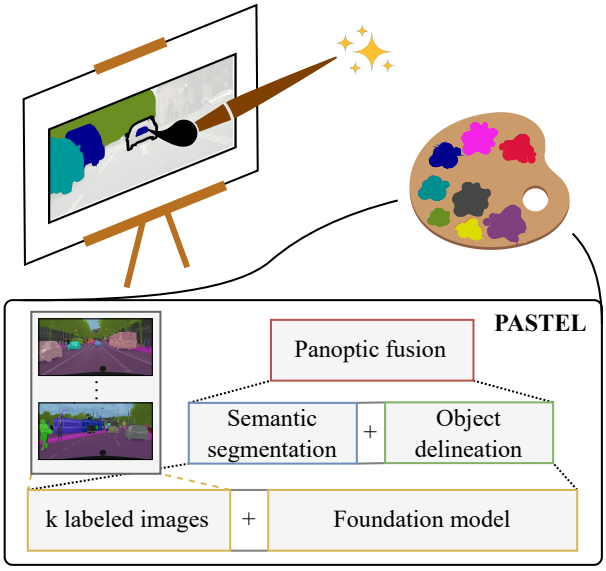

Fig. 1. We propose PASTEL for label-efficient panoptic segmentation. Our method combines a DINOv2 [15] backbone, creating descriptive image features, with labels from only k images, e.g., $k = 10$ on Citycapes [4]. A novel fusion module then merges semantic predictions with estimated object boundaries to yield the panoptic output.

In this work, we employ this paradigm shift to panoptic segmentation to substantially reduce the number of annotated images required for training. In particular, we propose a novel approach for *PAnoptic SegmenTation with fEw Labels* (PASTEL) and illustrate the key idea in Fig. 1. First, the vision foundation model DINOv2 [15] and a small set of k densely annotated images form the basis of PASTEL. Second, the descriptive image features of DINOv2 [15] allow for highly label-efficient training of two lightweight heads for semantic segmentation and object boundary estimation. Third, at inference time, a novel panoptic fusion module then merges the task-specific predictions and further refines their quality. Finally, PASTEL bootstraps selectively sampled unlabeled images for an additional performance boost via self-training. In extensive experiments, we demonstrate that PASTEL creates high-quality panoptic predictions from as few as 10 labeled images on Cityscapes [4], Pascal VOC [16], and PhenoBench [17]. Notably, PASTEL can hence be trained with labels produced by a single annotator in 1½ days [4] while

* Equal contribution.

This work was funded by the German Research Foundation Emmy Noether Program grant No 468878300.
[1] Niclas Vödisch, Kürsat Petek, Markus Käppeler, and Abhinav Valada are with the Department of Computer Science, University of Freiburg, Germany.
[2] Wolfram Burgard is with the Department of Engineering, University of Technology Nuremberg, Germany.
Digital Object Identifier (DOI): 10.1109/LRA.2024.3505779

outperforming previous label-efficient methods that require five to ten times as much data. We further show that the predictions of PASTEL can be used as pseudo-labels to train densely supervised models, i.e., rendering them label-efficient. To encourage future research, we release our code at http://pastel.cs.uni-freiburg.de.

II. Related Work

We provide an overview of visual foundation models and previous methods for label-efficient image segmentation.

Visual Foundation Models. The term "foundation model" defines models that are trained on large amounts of data for adaptation to a variety of downstream tasks [14]. First applied in natural language processing, e.g., GPT-3 [18], similar approaches have since also been proposed for computer vision (CV). For instance, CLIP [19] allows for zero-shot image classification that can be leveraged in open-vocabulary methods [20]. Florence [21] represents a general-purpose CV foundation model by extending the textual-visual shared representation to the space and time domains. Similarly, Painter [22] addresses common CV tasks such as image segmentation or depth estimation without task-specific heads. The recent SAM [23] enables zero-shot semantic and instance segmentation while lacking the ability to assign classes to the segmented areas. Finally, DINO [24] and DINOv2 [15] represent a new paradigm of visual foundation models relying on a completely unsupervised training scheme with neither cross-modal nor iterative human annotations. Nonetheless, these models have been shown to learn semantically descriptive features for downstream tasks [15], [13]. In this work, we exploit such image representations as a strong prior to enable label-efficient panoptic segmentation.

Label-Efficient Image Segmentation. Classical deep image segmentation methods require a large amount of annotated training data [1], [2], [3]. Therefore, many recent works employ different strategies of weak supervision [25] to reduce the labeling cost. For instance, unsupervised semantic segmentation is commonly addressed using contrastive learning techniques [10], [26] to find similar clusters in the feature space. Recent methods have leveraged descriptive image representations from large-scale task-agnostic pretraining [24] for both semantic [12] and instance segmentation [11]. With respect to the more challenging task of panoptic segmentation, CoDEPS [27] distills knowledge from a labeled source domain to a new unlabeled target domain. Sparse annotations offer an intermediate approach with limited pixel-based supervision generated by an inexpensive labeling scheme, e.g., point annotations [5], [6]. Semi-supervised training bootstraps a few densely annotated examples with a large set of unlabeled images and is commonly based on auxiliary tasks [7], self-training [8], or uncertainty estimation [9]. In this work, we add to the promising line of research that exploits visual foundation models for label-efficient dense supervision. The concurrent SPINO [13] approach combines a DINOv2 [15] backbone with separate heads for semantic segmentation and object boundary estimation. Inspired by these recent insights, we follow a similar design scheme but exploit further synergies between semantic classes and leverage unlabeled images via self-training. Importantly, unlike most prior label-efficient techniques, in this work we aim to enable panoptic segmentation from only as few labeled images as a single annotator can produce within a reasonable time frame, thus facilitating deployment in custom domains.

III. Technical Approach

In this section, we present our PASTEL approach for label-efficient panoptic segmentation including its network architecture, the training scheme, the novel panoptic fusion module, and the feature-driven iterative self-training.

A. Model Architecture and Training

The key insight of our PASTEL is to exploit the semantically rich image features from a foundation model to enable label-efficient segmentation and instance delineation.

Network Design. As illustrated in Fig. 2, we design our network according to the multi-task paradigm with a shared backbone. Inspired by the approaches Point2Mask [5] and SPINO [13], we separately perform pixel-based semantic segmentation and object boundary detection while using a shared backbone. In detail, we employ the pretrained ViT-B/14 variant of DINOv2 [15] as the frozen backbone. In the n-class segmentation head, we first upsample the patch-wise features of DINOv2 to the input image size, i.e., $14\times$-upsampling. We then feed the output to four 1×1 convolution layers of feature sizes 300, 300, 200, and n. In the object boundary head, we operate on a smaller feature map using a $4\times$-upsampling layer, again followed by four 1×1 convolution layers of output sizes 600, 600, 400, and 1. We frame the boundary detection task as binary classification with labels 0 and 1 denoting boundary and background pixels, respectively. During test-time, the output of both heads is merged by our novel panoptic fusion module as detailed in Sec. III-B.

Network Training. Due to the descriptive image features of the DINOv2 [15] backbone, we can train both heads with a minimum number k of annotated images. In practice, k can be as small as ten samples as shown in Sec. IV. To train the semantic segmentation head, we employ the bootstrapped cross-entropy loss [28] to compensate for an imbalanced class distribution:

$$\mathcal{L}_{sem} = \frac{-1}{K} \sum_{i=1}^{N} \mathbb{1}\left[p_{i,y_i} < t_K\right] \cdot \log(p_{i,y_i}), \tag{1}$$

where N denotes the number of pixels. Furthermore, p_{i,y_i} refers to the posterior probability of pixel i for the true class $y_i \in \{1, \dots, n\}$. Note that n corresponds to the number of semantic classes. The indicator function $\mathbb{1}(\cdot)$ returns 1 if p_{i,y_i} is smaller than the threshold t_K and 0 otherwise. To bootstrap pixels with yet uncertain predictions, i.e., a high loss, we set $t_K = 0.2$. Since we formulate the boundary detection as a 2-class classification task, we supervise this head with the binary cross entropy loss:

$$\mathcal{L}_{bnd} = \frac{-1}{N} \sum_{i=1}^{N} y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i), \tag{2}$$

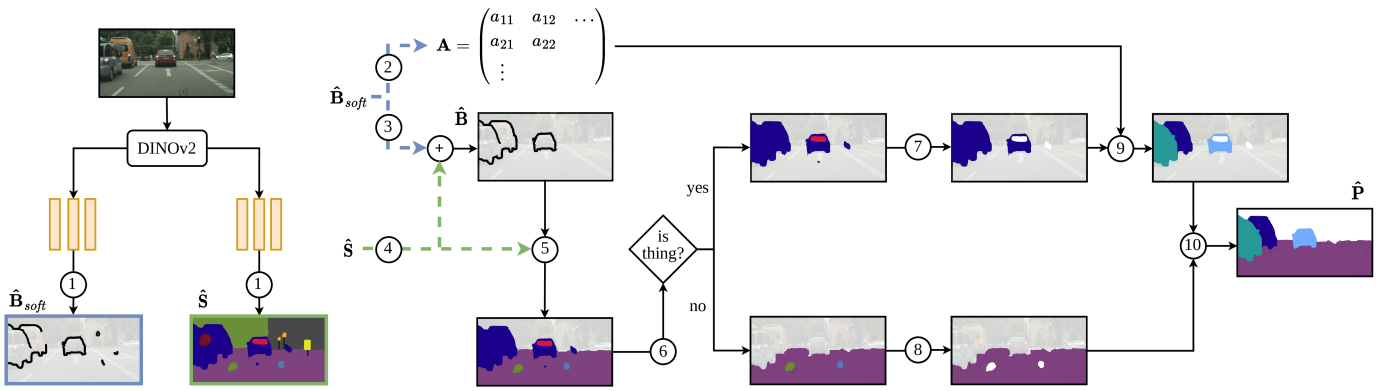

Fig. 2. Test-time overview of PASTEL illustrating the panoptic fusion scheme. For simplicity, we focus on *car* and *road* classes after step (1). The overall module is comprised of the following steps: (1) Overlapping multi-scale predictions; (2) Conversion of soft boundary map to an affinity matrix; (3) Boundary denoising; (4) Extraction of "stuff" to "thing" boundaries; (5) Class majority voting within enclosed areas; (6) Connected component analysis (CCA); (7) Filters on "thing" classes; (8) Filters on "stuff" classes; (9) Recursive two-way normalized cut (NCut) to separate connected instances; (10) Nearest neighbors-based hole filling of pixels with the *ignore* class.

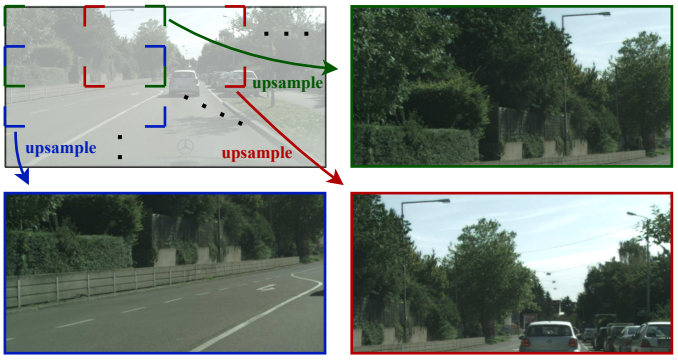

Fig. 3. We perform multi-scale test-time augmentation with overlapping image crops to mitigate visual artifacts at the borders. Before feeding the crops to the task-specific networks, we upsample them to the original image size. In this figure, we illustrate the approach for scale $s = 2$ and an image crop overlap of $z = 2$.

where N is the number of pixels, $y_i \in \{0, 1\}$ denotes the binary boundary label, and p_i refers to the pixel probability of being a boundary. During training, we set the true y_i to 0 if the instance identifier of a "thing" pixel differs from the identifier of any of its eight neighbors. Otherwise, we assign 1. If the pixel i belongs to a "stuff" class, we set $y_i = 1$.

In order to increase the variety of the small training set of only k samples, we employ extensive data augmentation. In particular, we perform randomized horizontal flipping and cropping with consecutive resizing to the input image size. We further augment various visual properties including brightness, contrast, saturation, and hue value.

B. Panoptic Fusion Module

Our proposed panoptic fusion module comprises three key steps: generating multi-scale predictions, a variety of heuristic-driven refinements, and the final instance delineation. We illustrate the overall methodology in Fig. 2.

Multi-Scale Prediction. During test-time, we perform both semantic segmentation and object boundary detection on multiple scales enabling our method to create more fine-grained predictions. In particular, we partition the input image of size (w, h) into smaller areas of size

$$w_s = w/s\,, \quad h_s = h/s\,, \tag{3}$$

where s denotes the scale. Importantly, we propose to utilize overlapping image crops with strides

$$r_{w,s} = \frac{w_s}{z}\,, \quad r_{h,s} = \frac{h_s}{z}\,. \tag{4}$$

Unlike non-overlapping ensembles [13], our approach prevents sharp borders within the merged prediction that can result in visual artifacts. The parameter z defines the extent of the overlap, e.g., $z = 2$ indicates that half of an image crop is overlapped by another crop. We depict this method in Fig. 3 for scale $s = 2$ and overlap $z = 2$, yielding nine image crops. We upsample each crop to the input image size (w, h) using bilinear interpolation and feed them through the respective head. Then, we downsample the generated feature maps to (w_s, h_s) and place them in a combined feature map at the position corresponding to the input image crop. We repeat this procedure for each scale and average features of overlapping pixels. Finally, we merge the features from multiple scales using the mean value per pixel.

Panoptic Fusion and Refinement. We visualize the individual steps of our proposed panoptic fusion module in Fig. 2, starting with the previously described multi-scale prediction (1). First, we compute an affinity matrix $\mathbf{A}$ from the predicted soft boundary $\hat{\mathbf{B}}_{soft}$ (2) to be used for instance delineation (9). We detail these steps in the next paragraph. In step (3), we obtain the binary boundary $\hat{\mathbf{B}}$ after thresholding the class probabilities of each element $b_{ij}^{soft} \in \hat{\mathbf{B}}_{soft}$ with λ_b:

$$b_{ij} = \begin{cases} 1 & \text{if } b_{ij}^{soft} > \lambda_b \\ 0 & \text{otherwise} \end{cases}\,, \; b_{ij} \in \hat{\mathbf{B}} \tag{5}$$

We further denoise $\hat{\mathbf{B}}$ by removing small boundaries. Next, we extract boundaries between any two "stuff" and "thing" classes (4) and add them to $\hat{\mathbf{B}}$. This enables us to find disconnected segments in the predicted semantic map $\hat{\mathbf{S}}$. In detail, in step (5) we perform connected component analysis (CCA) on $\hat{\mathbf{B}}$, followed by majority voting to update the pixel-wise predicted semantic classes $\hat{y}_i$ of a segment *seg*:

$$\hat{y}_i = \underset{y \in \{1,\dots,n\}}{\arg\max} \sum_{\hat{y}_j \in seg} \mathbb{1}\left[\hat{y}_j = y\right] \tag{6}$$

In Fig. 2, this changes the burgundy colored pixels in the left vehicles to the *car* class. In step (6), we again perform CCA

but use the semantic predictions, i.e., we obtain a segment for each area in the image whose neighbors belong to a different semantic category. In the following, we separate these segments into "stuff" and "thing" segments. First, we iterate over the "thing" segments (7) and set the semantic label to the *ignore* class if the segment size is below a threshold, the boundary head does not predict a boundary for any of the segments' pixels, or the segment is fully surrounded by another thing class. While the second filter is inspired by ensemble learning, the third filter targets infeasible objects *flying* in the scene. In Fig. 2, these filters remove the red pixels in the rear window of the center vehicle as well as the smaller *car* segments. Then, we iterate over the "stuff" segments (8) and apply the same filters except for the boundary-based removal. In step (9), we fuse the semantic prediction with the detected object boundaries to obtain a panoptic map. Finally, in step (10), we propagate labels from the nearest neighbor to fill all previously created holes, i.e., pixels that were assigned the *ignore* label. Note that although we only visualize *car* and *road* classes in Fig. 2, the final panoptic segmentation map $\hat{\mathbf{P}}$ contains all valid categories.

Instance Separation. Here, we further detail steps (2) and (9) as numbered in Fig. 2. While CCA on the pixels assigned to the same "thing" class already yields disconnected image segments, the number and exact location of instances within a segment remains unknown. To delineate instances, we employ recursive two-way normalized cut (NCut) [29] to each image segment of a "thing" class.

In step (2), we first downsample the soft boundary map $\hat{\mathbf{B}}_{soft}$ to size (w_b, h_b). Then, we compute a sparse affinity matrix $\mathbf{A} \in \mathbb{R}^{h_b \cdot w_b \times h_b \cdot w_b}$ based on the distance matrix $\mathbf{D}$with distances between pixels p_i and p_j defined as:

$$d_{ij} = \max_{p_l \in \text{line}(p_i, p_j)} \hat{\mathbf{B}}_{soft}(p_l)\,, \tag{7}$$

where where the $\text{line}(\cdot,\cdot)$ operator is provided by the Bresenham algorithm. We convert the distances to affinities by taking the negative exponential:

$$a_{ij} = e^{-\beta d_{ij}}\,, \tag{8}$$

where $a_{ij} \in \mathbf{A}$ and $d_{ij} \in \mathbf{D}$. The decay rate parameter β controls the sensitivity of the affinity to changes in the distance. We interpret $\mathbf{A}$ as a weighted radius neighborhood graph with nodes and edges representing image pixels and affinities between neighboring pixels, respectively.

In step (9), we mask those elements in $\mathbf{A}$ that are not part of the current image segment and apply recursive NCut to cut the segment into instances. The objective of NCut is to minimize the cost of dividing a graph into two separate sub-graphs. The continuous solution of NCut is given by the eigenvector v that corresponds to the second smallest eigenvalue λ of the generalized eigenvalue problem:

$$(\mathbf{C} - \mathbf{A})x = \lambda \mathbf{C} x\,, \tag{9}$$

where $\mathbf{C}$ is a diagonal matrix with diagonal elements $c_{ii} = \sum_j \mathbf{A}_{ij}$. The solution v is a continuous bipartition of the segment. If the cost of a cut, represented by λ, is less than a threshold and a stability criterion is fulfilled, we apply the cut. Otherwise, we stop the recursion. The stability criterion [29] measures the degree of smoothness in v. Formally, if the ratio between the minimum and the maximum value of a histogram of v is less than a threshold, the criterion holds. Thus, we do not cut if there is high uncertainty in the solution v. To finally apply the cut, we search for the splitting point that minimizes the NCut cost to bipartition v into two segments. We then recursively apply this procedure to both segments to find additional instances. If the size of a segment is below a threshold, we remove this segment and set its semantic label to the *ignore* class. While using CCA [13] fails for non-closed object boundaries, NCut is robust to small gaps and noise present in the boundary map.

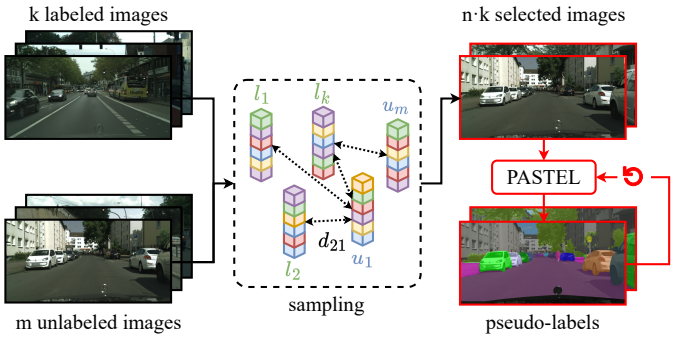

Fig. 4. During self-training, we extract feature vectors $\{l_1, l_2, \ldots, l_k\}$ of the labeled images as well as feature vectors $\{u_1, \ldots, u_m\}$ of unlabeled images. Since the performance of PASTEL is better on those unlabeled images that are more similar to the samples in the training set, we leverage the cosine similarity as distance measure d_{ij} for image sampling.

C. Iterative Self-Training

Due to leveraging multi-scale predictions followed by refinements made by our panoptic fusion module, the final output of PASTEL is of higher quality compared to the initial single-scale predictions, thus motivating iterative self-training. Because of the design of the fusion module, this enhancement mostly applies to the semantic output and is negligible for the boundary prediction. Therefore, we employ self-training only for the semantic segmentation head.

In particular, we propose to select unlabeled images in a feature-driven manner, as illustrated in Fig. 4. First, we generate feature representations of the selected k images with ground truth annotations as well as of a set of m unlabeled images using the DINOv2 [15] backbone. Then, we query the n nearest neighbors from the unlabeled set for each image in our training set. Inspired by place recognition in visual SLAM [30], we utilize the cosine similarity between feature vectors as a similarity measure for images.

$$\text{sim}_{\cos}(\mathbf{I_a}, \mathbf{I_b}) = \cos\left(\text{feat}(\mathbf{I_a}), \text{feat}(\mathbf{I_b})\right)\,, \tag{10}$$

where $\mathbf{I}$ denotes an image with corresponding features $\text{feat}(\mathbf{I})$. We observe that the semantic predictions for these similar images are better than the predictions of a randomly sampled image and can hence bootstrap the semantic head. Please see Sec. IV-C for a quantitative argument.

Next, we use PASTEL to create panoptic predictions for the sampled $n \cdot k$ images and treat them as pseudo-labels. We continue the training of the semantic segmentation head by constructing batches that contain both a ground truth annotation and a pseudo-labeled image. We use the same loss

TABLE I
IMAGE SEGMENTATION ON CITYSCAPES

Method	Backbone	Supervision	mIoU	PQ
1a) Mask2Former [2]	Swin-L	$\mathcal{L}$	82.9	66.6
2a) PiCIE† [10]	ResNet-18	$\mathcal{U}$	13.8	–
2b) STEGO† [12]	DINO	$\mathcal{U}$	38.0	–
3a) ST++ [8]	ResNet-50	$\mathcal{L}_{100} + \mathcal{U}$	61.4	–
4a) ST++ [8]	ResNet-50	$\mathcal{L}_{100}$	55.1	–
4b) Hoyer *et al.* [7]	ResNet-101	$\mathcal{L}^*_{100}$	62.1	–
4c) Hoyer *et al.*‡	DINOv2 ViT-B	$\mathcal{L}_{10}$	46.4	–
4d) ST++‡	DINOv2 ViT-B	$\mathcal{L}_{10}$	53.0	–
4e) PanopticDeepLab† [1]	DINOv2 ViT-B	$\mathcal{L}_{10}$	49.4	20.6
4f) Mask2Former‡	Swin-L	$\mathcal{L}_{10}$	50.7	29.2
4g) SPINO [13]	DINOv2 ViT-B	$\mathcal{L}_{10}$	61.2	36.5
5a) PASTEL (*ours*)	DINOv2 ViT-L	$\mathcal{L}_{100}$	75.5	50.7
5b) PASTEL (*ours*)	DINOv2 ViT-S	$\mathcal{L}^*_{100}$	64.2	41.0
5c) PASTEL (*ours*)	DINOv2 ViT-B	$\mathcal{L}_{10}$	63.3	41.3
5d) PASTEL (*ours*)	DINOv2 ViT-B	$\mathcal{L}_{10} + \mathcal{U}_{50}$	64.8	42.4

Supervision methods $\mathcal{L}$ and $\mathcal{U}$ denote labeled and unlabeled data. If a subscript k is specified, only k images were used for training. The metrics of $\mathcal{L}^*_{100}$ are averaged over the same three fixed sets [7]. †: Values are taken from SPINO [13]. ‡: Baselines trained by us.

TABLE II
IMAGE SEGMENTATION ON PASCAL VOC 2012

Method	Backbone	Supervision	mIoU	PQ
1a) Panoptic FCN [6]	ResNet-50	$\mathcal{L}$	80.2	67.9
2a) MaskContrast [26]	ResNet-50	$\mathcal{U}$	35.0	–
2b) MaskDistill [32]	ResNet-50	$\mathcal{U}$	48.9	–
3a) U^{2}PL [9]	ResNet-101	$\mathcal{L}_{92} + \mathcal{U}$	68.0	–
3c) ST++ [8]	ResNet-50	$\mathcal{L}_{92} + \mathcal{U}$	65.2	–
4a) U^{2}PL [9]	ResNet-101	$\mathcal{L}_{92}$	45.8	–
4b) ST++ [8]	ResNet-50	$\mathcal{L}_{92}$	50.7	–
4c) ST++‡	DINOv2 ViT-B	$\mathcal{L}_{20}$	52.8	–
5a) PASTEL (*ours*)	DINOv2 ViT-L	$\mathcal{L}_{92}$	71.1	47.3
5b) PASTEL (*ours*)	DINOv2 ViT-B	$\mathcal{L}_{20}$	60.6	37.0
5c) PASTEL (*ours*)	DINOv2 ViT-B	$\mathcal{L}_{20} + \mathcal{U}_{100}$	62.5	39.5

Supervision methods $\mathcal{L}$ and $\mathcal{U}$ denote labeled and unlabeled data, respectively. If a subscript k is specified, only k images were used for training. ‡: Baseline trained by us.

TABLE III
PHENOBENCH LEAF INSTANCE SEGMENTATION

Method	Backbone	Supervision	PQ (`val`)	PQ (`test`)
1a) Mask R-CNN [33]	ResNet-50	$\mathcal{L}$	61.5	59.7
2a) Mask R-CNN‡	ResNet-50	$\mathcal{L}_{15}$	41.5	38.2
2b) PASTEL (*ours*)	DINOv2 ViT-B	$\mathcal{L}_{15}$	51.7	49.0

In 2a) and 2b), we used 15 images for training. ‡: Baseline trained by us.

as in Eq. (1) for the ground truth sample but set $t_K = 1.0$ for the pseudo-labeled image.

IV. EXPERIMENTAL EVALUATION

We demonstrate that our PASTEL method outperforms previous label-efficient segmentation techniques while requiring significantly fewer annotations. We further showcase that using PASTEL as a plugin can render state-of-the-art segmentation models label-efficient. In extensive ablation studies, we analyze the various design choices.

A. Datasets and Implementation Details

We present results on three diverse datasets: First, the Cityscapes dataset [4] provides RGB images and high-quality panoptic annotations with 19 classes for urban driving. Second, the Pascal VOC 2012 dataset [16] was originally proposed as an object detection benchmark and has been substantially extended by SBD [31]. Concerning panoptic segmentation, the dataset comprises 20 "thing" classes and a single "stuff" class representing the background. Finally, the PhenoBench dataset [17] comprises several segmentation tasks for the agricultural domain. We apply our method to the leaf instance segmentation challenge. In stark contrast to autonomous driving, agricultural robotics lacks large-scale datasets underlining the importance of highly label-efficient approaches. In our experiments, we select k images from the `train` split of the respective dataset and report results on the `val` split. On PhenoBench, we provide further metrics for the `test` split. If not noted otherwise, we train both heads for 150 epochs on an Nvidia RTX A6000 GPU taking $11.5\,\mathrm{min}$ and $16.7\,\mathrm{min}$ for semantic segmentation and object boundary estimation, respectively. The inference time with our default settings is approx. $200\,\mathrm{s}$ per image. We discuss real-time deployment in the last paragraph of Sec. IV-B.

B. Panoptic Segmentation Results

For evaluation of PASTEL and other baseline methods, we report mIoU and PQ metrics [34]. While the mIoU only refers to semantic segmentation, the PQ measures the quality of performing panoptic segmentation.

Comparison With Related Works. We evaluate PASTEL with respect to previous label-efficient methods for semantic and panoptic segmentation. Additionally, we list metrics of fully supervised state-of-the-art approaches for panoptic segmentation on the respective dataset. Importantly, most prior techniques consider a minimum of 100 labeled images on Cityscapes and 92 on Pascal VOC, which is ten and five times more than our intended use case. Therefore, we report results for three different scenarios: First, with a minimum number of annotated images showcasing the label efficiency of our method. Second, we extend the setting to semi-supervision by including a few unlabeled images via our proposed self-training scheme. Finally, based on the encouraging results in Sec. IV-C, we investigate the potential performance of PASTEL when increasing the available compute resources and the number of annotated images, resembling the setup from previous label-efficient methods.

In Tab. I, we report results for the Cityscapes dataset. We first address the main target use case by allowing only ten annotated images for training. Our PASTEL (5c) achieves remarkable $63.3\,\%$ mIoU and $41.3\,\%$ PQ corresponding to an increase of $+2.1$ mIoU and $+4.8$ PQ to the recent SPINO [13] (4g). Notably, the state-of-the-art works Mask2Former [2] (4f) and PanopticDeepLab [1] adapted with a DINOv2 [15] backbone (4e) perform significantly worse revealing the need for specifically designed methods for extreme label efficiency. We further evaluate the work from Hoyer *et al.* [7] (4c) and ST++ [8] (4d) but replace the backbones with DINOv2 ViT-B to eliminate their impact. Besides the lack of instance predictions, this results in -16.9 and -10.3 mIoU compared to PASTEL. After employing our proposed self-training strategy (5d), the improvement over the concurrent SPINO [13] further

Fig. 5. We provide qualitative results on both Cityscapes (*left*) and Pascal VOC (*right*) for examples taken from the respective `val` split. The depicted results are generated by PASTEL based on the semi-supervised setup, i.e., $\mathcal{L}_k + \mathcal{U}_{n \cdot k}$.

Fig. 6. Qualitative results for the PhenoBench leaf instance segmentation challenge including different growth stages of the crops.

increases to $+3.6$ mIoU and $+5.9$ PQ, also yielding higher metrics than the semi-supervised ST++ [8] (3a) that is trained with more labels. For the third case, we use PASTEL with a DINOv2 ViT-S backbone (86M param.) (5b) and compare it with the ResNet-101-based [35] Hoyer *et al.* [7] (45M param.) (4b) when training on the same 100 annotations. Our approach yields $+2.1$ mIoU plus instance predictions. Finally, we show the potential of PASTEL with a DINOv2 ViT-L backbone (5a) that expands the increase to $+13.4$ mIoU. Notably, this reduces the gap to the fully supervised Mask2Fomer [2] (1a) to 7.4 mIoU and 15.9 PQ while using $3.4\,\%$ of the labels.

In Tab. II, we repeat similar experiments on the Pascal VOC dataset. When using only 20 annotated images, PASTEL (5b) yields $60.6\,\%$ mIoU and $37.0\,\%$ PQ, outperforming previous densely supervised methods with limited samples. Compared to ST++ [8] with a DINOv2 ViT-B backbone (4c), the metrics represent $+7.8$ mIoU when trained with the same images. Similar to Cityscapes, self-training further increases the performance of PASTEL (5c). For the third setup, PASTEL (5a) achieves an increase of $+20.4$ mIoU versus the supervision-only baseline reported in ST++ [8] (4b) when using the same number of annotations. On Pascal VOC, we can reduce the gap to the fully supervised Panoptic FCN [6] (1a) to 9.1 mIoU and 20.6 PQ while using only $0.8\,\%$ of the labels.

In Tab. III, we report results for the PhenoBench leaf instance segmentation task. Separating leaves is essential to assess the growth stage of crops and to detect diseases. However, annotating a conventional training set with hundreds of images is infeasible. Thus, we demonstrate that PASTEL (2b) significantly outperforms the best-performing baseline Mask R-CNN [33] from the dataset's benchmark, when only $1.1\,\%$ of the labeled images are available (2a).

TABLE IV
EVALUATION OF MASK2FORMER WITH RESNET-50

Training data	**Split**	mIoU	PQ	SQ	RQ
Ground truth	`train`	75.2	59.2	81.0	71.9
Pseudo-labels	`train`	63.2	44.0	76.5	54.6
Pseudo-labels	`train_extra`	64.6	44.8	76.5	55.8

Finally, we provide qualitative results in Fig. 5 and Fig. 6. Albeit the complexity of the Cityscapes scenes, we observe that PASTEL segments most *car* instances correctly. Further, the more challenging *pedestrians* are generally assigned the correct semantic class with minor over-segmentation of smaller body parts. The images of Pascal VOC are usually less complex and contain a smaller variety of classes within a single image. In the depicted results, PASTEL successfully separates instances of the same semantic class even in difficult scenes. For the PhenoBench leaf instance segmentation challenge, the predictions of PASTEL remain stable over the different growth stages of the crops.

Usage as Pseudo-Label Generator. In this experiment, we leverage the panoptic predictions of PASTEL as pseudo-labels to train a densely supervised panoptic segmentation model. In detail, we train Mask2Former [2] with a ResNet-50 [35] backbone using the official code for three different settings. First, we use the ground truth annotations of the `train` split of Cityscapes. Second, we generate panoptic pseudo-labels for the same data using PASTEL with the $\mathcal{L}_{10} + \mathcal{U}_{50}$ setting. Finally, we add pseudo-labels on the `train_extra` split showing how to leverage large unlabeled datasets with our method. On all pseudo-labels, we mask the static hood

TABLE V
COMPONENTS ANALYSIS

Method	mIoU	PQ	SQ	RQ
Scale 1 w. CCA inst. segm.	57.3	30.0	70.7	38.8
+ Multi-scale augmentation	62.4	36.8	73.9	46.8
+ Normalized cut	62.7	38.5	73.9	49.0
+ Refinement steps	63.3	41.3	74.5	52.1
+ Self-training (1 iteration)	**64.8**	**42.4**	**75.7**	**53.1**

Each row also includes the components of all rows above.

TABLE VI
NUMBER AND SELECTION OF LABELS

Count	mIoU	PQ	SQ	RQ
5	57.2	36.6	69.4	46.3
10	63.3	41.3	74.5	52.1
25	67.1	43.9	75.6	55.1
50	69.2	46.0	76.3	57.6
100	**70.7**	**47.2**	**76.7**	**59.0**
10 (*study*)	64.0±3.3	40.8±1.4	74.3±1.7	51.6±2.1

of the ego vehicle following previous works [36], [13]. We report the performance on `val` data in Tab. IV. Note that the numbers from the authors are slightly greater than our reproduced results, $+2.3$ mIoU and $+2.9$ PQ [2], but do not include SQ and RQ metrics. Importantly, the panoptic metrics with `train` pseudo-labels exceed the results obtained directly with PASTEL, i.e., training Mask2Former further bootstraps the panoptic segmentation performance without increasing the utilized number of human annotations. When adding the `train_extra` pseudo-labels, the panoptic segmentation scores can be further improved, achieving $+2.4$ PQ compared to the results of PASTEL. In summary, this experiment not only demonstrates that PASTEL can serve as a plugin rendering existing densely supervised segmentation models label-efficient but also enables real-time inference [2].

C. Ablations and Analysis

We conduct extensive ablation studies on Cityscapes [4] to analyze the effect of various components and hyperparameters. Throughout the tables, we highlight the parameters used in Sec. IV-B in gray. Except for the components analysis and the study on iterative self-training, we omit self-training to isolate the effect of a parameter. For further studies, e.g., image size, please refer to the supplementary material.

Components Analysis. We report the impact of the components of PASTEL in Tab. V. The largest effect can be observed for multi-scale test-time augmentation, enabling our method to produce more detailed predictions. Next, we substitute instance delineation via CCA [13] with recursive two-way normalized cut, improving the panoptic metrics. Employing the post-processing of our proposed panoptic fusion module increases both semantic and panoptic performance. Finally, we demonstrate that one iteration of self-training further boosts the performance by $+1.5$ mIoU and $+1.1$ PQ. In Tab. IX, we further show that our proposed feature-driven similarity sampling performs significantly better than self-training with 50 randomly sampled images.

Choice of Labeled Images. To measure the effect of the selected images, we conduct a user study with four participants tasked to select ten RGB images covering all semantic classes while maximizing diversity. We report the mean and standard deviation in the bottom row of Tab. VI denoted by *study*. The study shows that the performance of PASTEL remains stable for different selections of training data. Next, we evaluate the potential performance of PASTEL if one would further increase the number of labeled training images, although this does not reflect the main goal of our work. Please note that we use the same images as SPINO [13], allowing for a direct comparison. In detail, PASTEL achieves $+4.5$, $+5.4$, $+4.3$, $+5.1$, and $+4.3$ PQ compared to SPINO for an increasing label count from L_5 to L_{100}.

TABLE VII
VARIANTS OF THE BACKBONE

DINOv2	mIoU	PQ	SQ	RQ
ViT-S/14	53.2	34.1	73.9	42.6
ViT-B/14	63.3	41.3	74.5	52.1
ViT-L/14	**66.2**	44.2	75.0	55.9
ViT-g/14	65.8	**44.7**	**75.2**	**56.2**

TABLE VIII
NUMBER OF SELF-TRAINING ITERATIONS

Iterations	mIoU	PQ	SQ	RQ
0	63.3	41.3	74.5	52.1
1	**64.8**	**42.4**	**75.7**	**53.1**
2	62.9	41.4	75.2	51.8

We used 50 images and 50 epochs.

Backbone. We present results for different variants of DINOv2 [15] in Tab. VII. Note that we selected DINOv2 ViT-B/14 for our method as it compromises performance and computational feasibility. We observe that the larger backbone DINOv2 ViT-L/14 shows significant performance improvements, whereas increases due to DINOv2 ViT-g/14 are marginal. Similar to a study on image classification [15], we hypothesize that the number of parameters of the ViT-L/14 variant suffices to model the training data.

Iterative Self-Training. Finally, we conduct studies on the number of self-training iterations (Tab. VIII) and images sampled by our sampling strategy (Tab. IX). We show that performing self-training once is sufficient to increase performance [37], while further iterations decrease the quality, most likely due to overfitting. For the image count, we sample the $n = 5$ nearest neighbors for each annotated image in the training set and continue training the semantic head for 50 epochs. In Tab. IX, we further report results for $n \in \{0, 1, 10, 20\}$, where $n = 0$ corresponds to resuming the training without pseudo-labeled images. Note that our proposed similarity sampling yields significantly better results than self-training on randomly sampled images, shown in the bottom row.

V. CONCLUSION

In this work, we demonstrated that recent visual foundation models offer a powerful pretraining strategy for solving computer vision tasks in a label-efficient manner. In particular, we presented PASTEL for label-efficient panoptic segmentation. Our method combines descriptive image features from a DINOv2 [15] backbone with two lightweight heads for semantic segmentation and object boundary detection. It can be trained with as few as ten annotated images. We showed that our

TABLE IX
NUMBER OF SELF-TRAINING IMAGES

Count	mIoU	PQ	SQ	RQ
0	63.7	41.4	74.9	52.1
10	63.5	41.5	75.1	52.1
50	**64.8**	42.4	**75.7**	53.1
100	63.7	**42.6**	75.3	**53.4**
200	**64.8**	42.5	75.3	**53.4**
50 (*random*)	61.8±1.3	38.5±0.5	73.6±0.3	48.5±0.7

We used one iteration of self-training with 50 epochs. For the randomly sampled images, we provide mean and standard deviation over three experiments.

novel panoptic fusion module yields substantial performance improvements compared to previous works and illustrated how to further enhance the results using self-training with similar images. Most notably, we demonstrated that PASTEL sets the new state of the art for label-efficient segmentation by improving mIoU scores by $+13.4$ and $+20.4$ on Cityscapes and Pascal VOC datasets, respectively. In future research, we aim to further close the gap to fully supervised methods paving the way for widespread application of panoptic segmentation without requiring large-scale annotated datasets.

REFERENCES

[1] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, "Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation," in *Conf. on Comput. Vis. and Pattern Recog.*, 2020, pp. 12 472–12 482.

[2] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Conf. on Comput. Vis. and Pattern Recog.*, 2022, pp. 1280–1289.

[3] R. Mohan and A. Valada, "Perceiving the invisible: Proposal-free amodal panoptic segmentation," *Rob. and Autom. Letters*, vol. 7, no. 4, pp. 9302–9309, 2022.

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes dataset for semantic urban scene understanding," in *Conf. on Comput. Vis. and Pattern Recog.*, 2016, pp. 3213–3223.

[5] W. Li, Y. Yuan, S. Wang, J. Zhu, J. Li, J. Liu, and L. Zhang, "Point2Mask: Point-supervised panoptic segmentation via optimal transport," in *Int. Conf. on Comput. Vis.*, October 2023, pp. 572–581.

[6] Y. Li, H. Zhao, X. Qi, Y. Chen, L. Qi, L. Wang, Z. Li, J. Sun, and J. Jia, "Fully convolutional networks for panoptic segmentation with point-based supervision," *Trans. on Pattern Anal. and Mach. Intell.*, vol. 45, no. 4, pp. 4552–4568, 2023.

[7] L. Hoyer, D. Dai, Y. Chen, A. Köring, S. Saha, and L. Van Gool, "Three ways to improve semantic segmentation with self-supervised depth estimation," in *Conf. on Comput. Vis. and Pattern Recog.*, 2021, pp. 11 125–11 135.

[8] L. Yang, W. Zhuo, L. Qi, Y. Shi, and Y. Gao, "ST++: Make self-training work better for semi-supervised semantic segmentation," in *Conf. on Comput. Vis. and Pattern Recog.*, 2022, pp. 4258–4267.

[9] Y. Wang, H. Wang, Y. Shen, J. Fei, W. Li, G. Jin, L. Wu, R. Zhao, and X. Le, "Semi-supervised semantic segmentation using unreliable pseudo-labels," in *Conf. on Comput. Vis. and Pattern Recog.*, 2022, pp. 4238–4247.

[10] J. Hyun Cho, U. Mall, K. Bala, and B. Hariharan, "PiCIE: Unsupervised semantic segmentation using invariance and equivariance in clustering," in *Conf. on Comput. Vis. and Pattern Recog.*, 2021, pp. 16 789–16 799.

[11] X. Wang, R. Girdhar, S. X. Yu, and I. Misra, "Cut and learn for unsupervised object detection and instance segmentation," in *Conf. on Comput. Vis. and Pattern Recog.*, 2023, pp. 3124–3134.

[12] M. Hamilton, Z. Zhang, B. Hariharan, N. Snavely, and W. T. Freeman, "Unsupervised semantic segmentation by distilling feature correspondences," in *Int. Conf. on Learn. Represent.*, 2022.

[13] M. Käppeler, K. Petek, N. Vödisch, W. Burgard, and A. Valada, "Few-shot panoptic segmentation with foundation models," in *Intern. Conf. on Rob. and Autom.*, 2024, pp. 7718–7724.

[14] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[15] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, *et al.*, "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research*, 2024.

[16] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. of Comput. Vis.*, vol. 88, pp. 303–338, 2010.

[17] J. Weyler, F. Magistri, E. Marks, Y. L. Chong, M. Sodano, G. Roggiolani, N. Chebrolu, C. Stachniss, and J. Behley, "PhenoBench – A large dataset and benchmarks for semantic image interpretation in the agricultural domain," *Trans. on Pattern Anal. and Mach. Intell.*, vol. 46, no. 12, pp. 9583–9594, 2024.

[18] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, *et al.*, "Language models are few-shot learners," in *Conf. on Neural Inform. Process. Syst.*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, 2020, pp. 1877–1901.

[19] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, *et al.*, "Learning transferable visual models from natural language supervision," in *Conf. on Robot Learning*, vol. 139, 2021, pp. 8748–8763.

[20] Y. Lin, M. Chen, W. Wang, B. Wu, K. Li, B. Lin, H. Liu, and X. He, "CLIP is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation," in *Conf. on Comput. Vis. and Pattern Recog.*, 2023, pp. 15 305–15 314.

[21] L. Yuan, D. Chen, Y.-L. Chen, N. Codella, X. Dai, J. Gao, H. Hu, X. Huang, B. Li, C. Li, *et al.*, "Florence: A new foundation model for computer vision," *arXiv preprint arXiv:2111.11432*, 2021.

[22] X. Wang, W. Wang, Y. Cao, C. Shen, and T. Huang, "Images speak in images: A generalist painter for in-context visual learning," in *Conf. on Comput. Vis. and Pattern Recog.*, 2023, pp. 6830–6839.

[23] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," in *Int. Conf. on Comput. Vis.*, 2023, pp. 3992–4003.

[24] M. Caron, H. Touvron, I. Misra, H. Jegou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Int. Conf. on Comput. Vis.*, 2021, pp. 9630–9640.

[25] W. Shen, Z. Peng, X. Wang, H. Wang, J. Cen, D. Jiang, *et al.*, "A survey on label-efficient deep image segmentation: Bridging the gap between weak supervision and dense prediction," *Trans. on Pattern Anal. and Mach. Intell.*, vol. 45, no. 8, pp. 9284–9305, 2023.

[26] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, and L. Van Gool, "Unsupervised semantic segmentation by contrasting object mask proposals," in *Int. Conf. on Comput. Vis.*, 2021, pp. 10 032–10 042.

[27] N. Vödisch, K. Petek, W. Burgard, and A. Valada, "CoDEPS: Online continual learning for depth estimation and panoptic segmentation," *Robotics: Science and Systems*, 2023.

[28] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe, "Full-resolution residual networks for semantic segmentation in street scenes," in *Conf. on Comput. Vis. and Pattern Recog.*, 2017, pp. 3309–3318.

[29] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Trans. on Pattern Anal. and Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[30] N. Vödisch, D. Cattaneo, W. Burgard, and A. Valada, "Continual SLAM: Beyond lifelong simultaneous localization and mapping through continual learning," in *Robotics Research*, 2023, pp. 19–35.

[31] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *Int. Conf. on Comput. Vis.*, 2011, pp. 991–998.

[32] W. Van Gansbeke, S. Vandenhende, and L. Van Gool, "Discovering object masks with transformers for unsupervised semantic segmentation," *arXiv preprint arXiv:2206.06363*, 2022.

[33] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Int. Conf. on Comput. Vis.*, 2017, pp. 2980–2988.

[34] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Conf. on Comput. Vis. and Pattern Recog.*, 2019, pp. 9396–9405.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Conf. on Comput. Vis. and Pattern Recog.*, 2016, pp. 770–778.

[36] L.-C. Chen, R. G. Lopes, B. Cheng, M. D. Collins, E. D. Cubuk, B. Zoph, H. Adam, and J. Shlens, "Naive-Student: Leveraging semi-supervised learning in video sequences for urban scene segmentation," in *Eur. Conf. on Comput. Vis.*, 2020, pp. 695–714.

[37] X. Wang, Z. Yu, S. De Mello, J. Kautz, A. Anandkumar, C. Shen, and J. M. Alvarez, "FreeSOLO: Learning to segment objects without annotations," in *Conf. on Comput. Vis. and Pattern Recog.*, 2022, pp. 14 156–14 166.

A Good Foundation is Worth Many Labels: Label-Efficient Panoptic Segmentation

- Supplementary Material -

Niclas Vödisch[1*], Kürsat Petek[1*], Markus Käppeler[1*], Abhinav Valada[1], and Wolfram Burgard[2]

In this supplementary material, we provide additional experiments, ablation studies, and qualitative results. We conclude by discussing some limitations of our proposed PASTEL approach.

S-I. Pseudo-Labels for Efficient Pretraining

In this section, we extend the evaluation of leveraging PASTEL as a pseudo-label generator to enable label-efficient training of any existing panoptic segmentation model. As we demonstrate in Sec. IV-B, this process upgrades existing models from the classical dense supervision style, which requires many annotated images, to label-efficient training. In Tab. S-I, we provide results from employing the predicted panoptic maps from both the `train` and `train_extra` splits of Cityscapes [4] as pseudo-labels to train Mask2Former [2] with a ResNet-50 [35] backbone. In contrast to Tab. IV, we interpret this step only as pretraining and resume the training with the ground truth annotations from the `train` set. In comparison to the results reported by the authors, who train only on the ground truth annotations, our label-efficient pretraining results in an increase of $+1.9$ mIoU and $+1.4$ PQ scores.

TABLE S-I
Evaluation of Mask2Former with ResNet-50

Label type	Data split	mIoU	PQ	SQ	RQ
Ground truth†	`train`	77.5	62.1	–	–
Pseudo-labels	`train_extra`	64.6	44.8	76.5	55.8
+ Ground truth	`train`	79.4 (+1.9)	63.5 (+1.4)	82.2	76.4

Supervised training results of Mask2Former [2] with a ResNet-50 [35] backbone. We pretrain the network on pseudo-labels generated by PASTEL and then continue training on ground truth annotations. The results denoted by † are reported by the authors [2].

S-II. Ablations and Analysis

In this supplementary section, we further extend the ablation studies provided in Sec. IV-C. We continue to highlight the parameters used in Sec. IV-B in gray. We further omit self-training to isolate the effect of the analyzed parameter.

Image Size. In this study, we ablate the effect of the image size on the overall performance. We report results in Tab. S-II for the full resolution as well as on scales $1/2$ and $1/4$. While scale $1/2$ achieves decent metrics, the performance significantly decreases for scale $1/4$.

Number of Epochs. In Tab. S-III, we evaluate the performance of PASTEL after different numbers of training epochs showing a general trend of improvements until the metrics converge. Please note that we use a loss-based termination strategy during training, i.e., we explicitly do not select the number of epochs with the highest metrics in Tab. S-III as we consider them to represent test data.

TABLE S-II
Image Size

Image size	mIoU	PQ	SQ	RQ
252×504	56.7	31.9	71.4	41.0
504×1008	**63.5**	39.9	73.2	50.8
1022×2044	63.3	**41.3**	**74.5**	**52.1**

TABLE S-III
Number of Epochs

Epoch	mIoU	PQ	SQ	RQ
50	60.5	39.5	74.6	49.7
100	63.4	41.0	74.8	51.8
150	63.3	41.3	74.5	**52.1**
200	**63.7**	**41.4**	**74.9**	**52.1**
250	63.0	41.0	75.1	51.5

S-III. Discussion of Limitations

Our label-efficient segmentation approach is subject to two limitations. First, since PASTEL predicts the boundaries of objects based on the RGB input, areas that are separated in the 2D image space but belong to the same real-world objects cannot be assigned the same instance ID. We visualize examples of this failure case for occluded objects in Fig. S-3. In the upper image, the car on the right is cut into two instances due to occlusion by a traffic light. In the lower image, PASTEL assigns four different instance IDs to different parts of the bus, which is occluded by multiple poles. Potential solutions to this limitation would be to employ amodal panoptic segmentation or a separate network head to predict the pixel offset similar to classical bottom-up methods. A key challenge of this future work will be to enable training with as few labeled images as utilized by PASTEL. Second, the minimum number of labeled images is constrained by the necessity for all considered classes to be present in the selected images. However, this constraint is not related to the proposed approach but is rather inherent in the need of the network to learn the notion of classes.

* Equal contribution.
[1] Department of Computer Science, University of Freiburg, Germany.
[2] Department of Eng., University of Technology Nuremberg, Germany.

Fig. S-1. Object boundaries predicted by PASTEL based on the semi-supervised setup, i.e., $\mathcal{L}_{10} + \mathcal{U}_{50}$.

Fig. S-2. Additional qualitative results on both Cityscapes (*left*) and Pascal VOC (*right*) datasets for examples taken from the respective `val` split. The depicted results are generated by PASTEL based on the semi-supervised setup, i.e., $\mathcal{L}_k + \mathcal{U}_{n \cdot k}$.

Fig. S-3. A limitation of our method is that occluded objects are separated into multiple instances.

S-IV. Qualitative Results

In Fig. S-2, we present additional qualitative results for both Cityscapes [4] and Pascal VOC [16] datasets. In Fig. S-4, we provide further qualitative results for the PhenoBench [17] leaf instance segmentation challenge.

Cityscapes. The examples from the Cityscapes dataset include scenes from all three cities within the `val` split, i.e., Frankfurt, Lindau, and Munster. Notably, our employed multi-scale prediction scheme allows for segmenting also more distanced details such as traffic signs. In Fig. S-1, we further visualize the predicted object boundary of the examples shown in Fig. 5 of the main paper. As we observe in the pedestrian scenes, the over-segmentation of small body parts is caused by the predicted boundaries.

Pascal VOC. Since the majority of images in the Pascal VOC dataset contain only a single object, we deliberately show results on scenes with multiple objects including multiple instances of the same class as well as compositions of different "thing" classes.

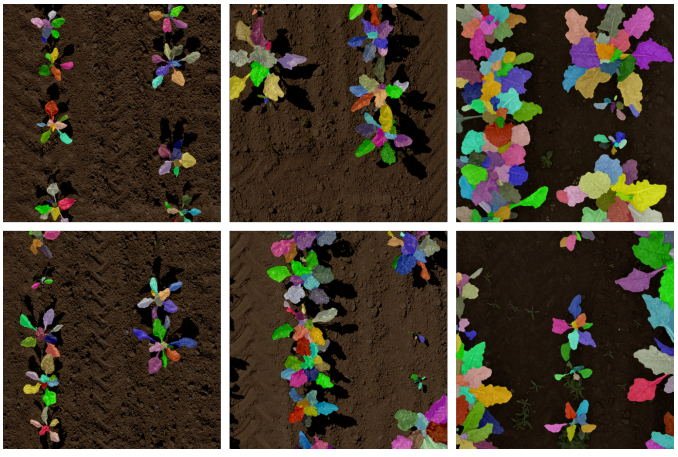

Fig. S-4. Additional qualitative results for the PhenoBench leaf instance segmentation challenge including different growth stages of the crops.

PhenoBench. Separating leaves is an important task for estimating the growth stage of plants and for detecting leaf diseases [17]. We provide results for all three stages contained in the `val` split of the dataset. Despite a drastic increase in scene complexity due to overlapping leaves, the prediction quality remains stable across the different stages.