Panoptic Segmentation with Partial Annotations for Agricultural Robots

Jan Weyler Thomas Läbe Jens Behley Cyrill Stachniss

***Abstract*—A detailed analysis of agricultural fields is key toward reducing the use of agrochemicals to achieve a more sustainable crop production. To this end, agricultural robots equipped with vision-based systems offer the potential to detect individual plants in the field automatically. This capability enables targeted management actions in the field, effectively reducing the amount of agrochemicals. A primary target of such vision systems is to perform a panoptic segmentation, combining the task of semantic and instance segmentation. Recent methods use neural networks for this task, which typically have to be trained on densely annotated images containing the required ground truth information for each pixel. Gathering these dense annotations is generally daunting and requires domain experts' knowledge in the agricultural domain. In this paper, we propose a method to effectively reduce the annotation bottleneck and yet achieve high performance using partial annotations. These partial annotations contain ground truth information only for a subset of pixels per image and are thus much faster to obtain than dense annotations. We propose a novel set of losses that exploit measures from vector fields used in physics, i.e., divergence and curl, to effectively supervise predictions without ground truth annotations. The experimental evaluation shows that our approach outperforms several state-of-the-art methods targeting to reduce the amount of annotations.**

***Index Terms*—Robotics and Automation in Agriculture and Forestry, Semantic Scene Understanding, Deep Learning for Visual Perception**

I. INTRODUCTION

DECREASING the detrimental environmental impacts of agrochemicals such as herbicides and pesticides by reducing their application in conventional farming is crucial to achieve a more sustainable agriculture. Autonomous agricultural robots offer the potential to tackle this challenge [6]. Such platforms, equipped with vision-based systems, can employ plant classification systems [26] to perform targeted field interventions and reduce the use of agrochemicals [17].

Thus, a key objective is to develop vision-based models addressing the task of panoptic segmentation [12], which targets a joint semantic and instance segmentation [3], [18]. The former assigns each image pixel to the class background, crop, or weed. Contrary, the latter aims to generate pixel-precise binary masks for each plant instance. Thus, panoptic segmentation empowers agricultural robots to perform plant-specific treatments. Besides, for different tasks like vision-based navigation, it enables splitting the background into more classes that may occur in real fields. Most recent vision-based systems [22] deploy convolutional neural networks (CNNs) for this task which are trained on densely annotated images, providing ground truth information for all pixels, see Fig. 1. However, gathering these dense annotations is laborious and, at the same time, requires domain expertise to distinguish crops and weeds. Thus, collecting large, densely annotated datasets is often a bottleneck.

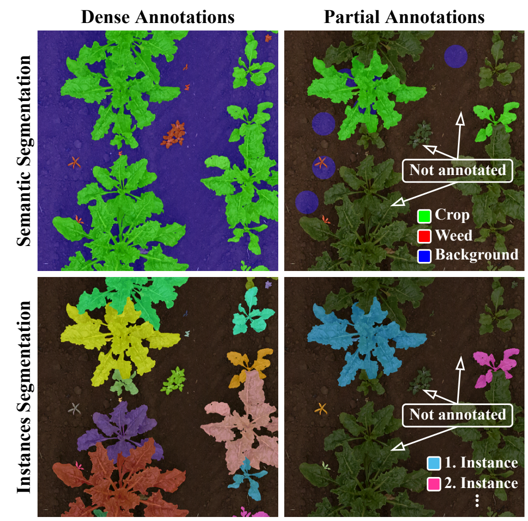

Fig. 1: We propose a vision-based approach based on CNNs to perform a panoptic segmentation, combining a semantic segmentation of background, crop, or weed and an instance segmentation of plants. Our model is trainable with partial annotations, contrasting with most other methods relying on dense annotations.

In this paper, we aim to overcome the bottleneck of requiring dense annotations for panoptic segmentation by proposing a network architecture trainable with substantially reduced annotations. Our method requires only annotations for a subset of plant instances per image, which we call *partial annotations*, see Fig. 1. Unlike dense annotations, partial annotations are easier to collect at a larger scale and facilitate the creation of a diverse training set by annotating a few plants in multiple images taken under varying conditions instead of annotating all plants within a single image.

The main contribution of this paper is a novel end-to-end trainable pipeline for panoptic segmentation targeting the annotation bottleneck. We propose a network architecture trainable with partial annotations, including three novel, conceptually simple, yet effective loss functions. These losses exploit concepts from vector calculus, specifically divergence, curl,

Manuscript received: Sep 8, 2023; Revised: Nov 7, 2023; Accepted: Dec 13, 2023. This paper was recommended for publication by Editor Hyungpil Moon upon evaluation of the Associate Editor and Reviewers' comments.

All authors are with the University of Bonn, Germany. Cyrill Stachniss is additionally with the Department of Engineering Science at the University of Oxford, UK, and with the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany.

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

and consistency of vector fields, to provide explicit supervision for pixels corresponding to instances without annotations during training. Ultimately, our proposed losses encourage the network to predict a vector field that enables us to recover individual instances through clustering. Additionally, each loss increases the supervision and performance of our model. Finally, we achieve superior performance to other methods targeting the annotation bottleneck. The source code of our method is available at: https://github.com/PRBonn/PSPA.

II. Related Work

Several vision-based approaches have been proposed for panoptic, semantic, and instance segmentation utilizing different annotation schemes for supervision. Below, we provide a broad overview of methods employing varying supervision.

Full Supervision. Most methods for panoptic [12], [22], semantic [15], [23], and instance [4], [7], [19] segmentation are fully supervised, i.e., they consume densely annotated training data providing for each pixel corresponding ground truth information. In the agricultural domain, Roggiolani et al. [18] present an approach for panoptic segmentation, which involves the classification of crops and weeds, while jointly identifying instances of plants and leaves. Furthermore, Milioto et al. [18] propose a method for semantic segmentation of crops and weeds that leverages background knowledge to speed up training and improve generalization capabilities. Similarly, McCool et al. [17] perform a segmentation of weeds on agricultural robots by using multiple lightweight CNNs in a mixture model. Regarding instance segmentation, Halstead et al. [6] propose a method based on Mask R-CNN [7] enabling crop-agnostic monitoring in arable farmlands. Similarly, Champ et al. [3] also utilize Mask R-CNN to detect and remove individual weeds in agricultural fields. Despite its need for densely annotated images, Mask R-CNN remains a prevalent choice in the agricultural domain. While it allows to ignore specific instances during training, such instances must be explicitly assigned to an ignore class, preventing any degradation of the training. This contrasts with our method operating without such extra information.

Weak Supervision. Recent methods for instance segmentation target to reduce the cost of dense pixel-wise annotations. Specifically, they require only box-level annotations for all instances during training while generating pixel-wise masks at inference [14], [25]. To this end, Hsu et al. [8] propose to exploit the bounding box tightness prior during training to supervise mask predictions without ground truth. Petti et al. [21] apply this method in the agricultural domain to count cotton flowers from aerial images. Another approach is to deploy promptable foundation models for image segmentation, e.g., the segment anything model (SAM) [13]. These large-scale models are pretrained on massive datasets and provide high-quality, class-agnostic instance masks based on prompts, e.g., bounding boxes. However, Ji et al. [9] show that the zero-shot capabilities of SAM are limited in the agricultural domain. Thus, a two-step, weakly-supervised approach [20] is more promising in the agricultural domain. Typically, an object detector like DETR [2] is first trained using box-level annotations. Next, during inference, the predicted bounding boxes are passed as prompts to SAM, resulting in a pixel-wise segmentation for each prompt. In contrast to our method, these approaches demand annotations for every instance within an image, even when only bounding boxes are needed.

Partial Supervision. Other recent approaches target to reduce the annotation bottleneck of instance segmentation differently. Specifically, these methods require complete annotations for a subset of instances within each image while leaving the remaining instances unannotated, i.e., partial annotations, see Fig. 1. With this aim, Wolny et al. [28] propose a class-agnostic instance segmentation approach based on non-spatial embeddings. They exploit a set of losses to structure the embedding space of annotated and unannotated instances such that individual instances can be recovered. In contrast to our approach, their proposed losses primarily promote consistency in the non-spatial embedding space, whereas we define multiple losses explicitly encouraging the network to predict spatial embeddings that are well-suited to recover individual instances through clustering. Additionally, our method is not class-agnostic, i.e., it can readily handle multiple classes, as required for our task. Kigli et al. [10] show that in the context of semantic segmentation, a reduced set of semantic annotations per image is sufficient to achieve solid results. However, their approach is restricted to semantic segmentation while we perform a panoptic segmentation with partial supervision.

III. Our Approach

Our main objective is to develop a CNN-based model that empowers agricultural robots to perform a panoptic segmentation in real agricultural fields. Specifically, our model is trainable with partial annotations, where only a subset of plant instances is annotated in each training image.

A. General Architectural Concept

For the task of panoptic segmentation, we employ a proposal-free encoder-decoder network that consists of three components. First, we utilize a shared encoder backbone ϕ_{enc} to generate a compact but expressive representation of an input image. Next, we pass this output to two decoupled decoders, providing task-specific dense predictions, see Fig. 2.

The first decoder ϕ_{sem} predicts the semantic segmentation by assigning each pixel to the class background, crop, or weed. In the context of panoptic segmentation, the background belongs to the *stuff* classes containing any object not belonging to vegetation, while crops and weeds belong to the *thing* classes [12]. However, our architecture is easily extendable to multiple *stuff* classes. The second decoder ϕ_{off} performs an instance segmentation based on the principle of spatial embeddings [19]. The output of this decoder is a 2D vector field providing for each pixel belonging to a specific plant an offset vector pointing toward its instance's centroid. Next, we obtain 2D spatial embeddings, where embeddings belonging to the same instance form a cluster by translating each pixel of a crop or weed along its predicted offset vector. Finally, we recover instances at inference by applying a clustering algorithm based on these spatial embeddings. Our pipeline enables

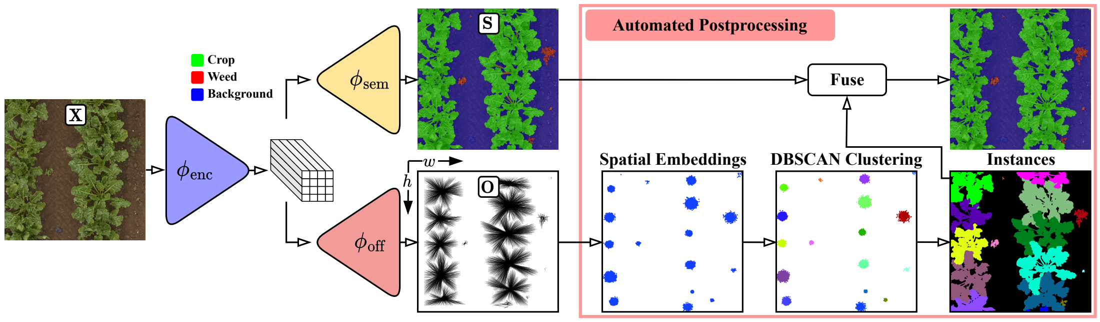

Fig. 2: Our network for panoptic segmentation based on an encoder-decoder architecture. First, we pass the input image to the encoder to generate a compact but expressive representation. Next, we pass this representation to the upper decoder to perform a semantic segmentation. Simultaneously, the lower decoder predicts a vector field, where offset vectors point toward their associated plant centroid. We exploit these offsets in an automated postprocessing procedure to compute spatial embeddings, which we cluster to recover individual instances. Finally, we fuse the instances and semantics such that each instance belongs to a single class.

training with either dense or partial annotations, allowing us to adapt the training process to different annotation schemes. We propose novel losses targeting partial annotations, enabling us to supervise offset vectors of instances without annotations. Next, we explain both setups in more detail, starting with dense annotations.

B. Training Dataset with Dense Annotations

In the fully supervised setting, we have access to a densely annotated training dataset $\mathcal{D} = \{(\mathbf{X}, \mathbf{Y}, \mathcal{C})\}$ containing N RGB images $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$, i.e., $|\mathcal{D}| = N$. Let H and W denote the height and width of an image. We define $\mathbf{Y} \in \mathbb{N}^{H \times W \times K}$ as the dense semantic annotation that contains for each pixel a one-hot encoded vector over K classes. In our work, we have $K = 3$ classes, i.e., background, crop, and weed. Each image also contains a set of M instances $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M\}$, where $\mathcal{C}_j$ is the subset of all pixel coordinates $\mathcal{X} = \{1, 2, \ldots, H\} \times \{1, 2, \ldots, W\}$ belonging to the j$^{\text{th}}$ instance, i.e., $\mathcal{C}_j \subset \mathcal{X}$.

C. Semantic Segmentation with Dense Annotations

For semantic segmentation, we train our network to generate a prediction $\mathbf{S} \in \mathbb{R}^{H \times W \times K}$ for each input image, representing a pixel-wise categorical distribution over K possible classes. Hence, we minimize during training the cross-entropy, comparing the prediction and ground truth class for each pixel independently:

$$\mathcal{L}_{\text{sem}}\left(\mathbf{Y}, \mathbf{S}, \mathcal{X}\right) = -\frac{1}{|\mathcal{X}|} \sum_{k=1}^{K} \sum_{(h,w) \in \mathcal{X}} \mathbf{Y}_{hwk} \log\left(\mathbf{S}_{hwk}\right), \quad (1)$$

where we employ the shared encoder and semantic-specific decoder to compute $\mathbf{S} = \phi_{\text{sem}}\left(\phi_{\text{enc}}\left(\mathbf{X}\right)\right)$.

At inference, we assign each pixel to the class with the highest confidence to obtain the final semantic segmentation.

D. Instance Segmentation with Dense Annotations

The aim of our proposed instance segmentation is to train the network to predict for a given input image a 2D vector field $\mathbf{O} \in \mathbb{R}^{H \times W \times 2}$. For each pixel coordinate $\mathbf{x}_i = [h_i, w_i]^\intercal \in \mathcal{X}$ associated with the j$^{\text{th}}$ instance, this vector field should contain a 2D offset vector $\mathbf{o}_i = \left[o_h\left(h_i, w_i\right), o_w\left(h_i, w_i\right)\right]^\intercal$ such that the resulting spatial embedding $\mathbf{e}_i = \mathbf{x}_i + \mathbf{o}_i$ is close to the embedding centroid $\mathbf{c}_j = \frac{1}{|\mathcal{C}_j|} \sum_{\mathbf{x}_i \in \mathcal{C}_j} \left(\mathbf{x}_i + \mathbf{o}_i\right)$ of this instance, as illustrated in Fig. 2. Here, we employ the shared encoder and instance-specific decoder to compute $\mathbf{O} = \phi_{\text{off}}\left(\phi_{\text{enc}}\left(\mathbf{X}\right)\right)$.

During training, we employ a clustering loss function [19] to optimize the model parameters toward obtaining the desired vector field. Crucially, we generate for each annotated instance $\mathcal{C}_j$ a soft mask based on the current offset predictions. Specifically, we use a function $f_{\mathcal{C}_j} : \mathbb{R} \to [0, 1]$ converting the distance between a spatial embedding $\mathbf{e}_i$ to the centroid $\mathbf{c}_j$ into a score of belonging to this instance:

$$f_{\mathcal{C}_j}\left(\mathbf{e}_i\right) = \exp\left(-\frac{\|\mathbf{e}_i - \mathbf{c}_j\|^2}{2\sigma^2}\right). \quad (2)$$

Hence, a high score implies the association of $\mathbf{e}_i$ with the j$^{\text{th}}$ instance. Conversely, a low score indicates its association with the background or another instance. Here, the hyperparameter σ defines an isotropic clustering region around the centroid. Finally, we obtain a soft mask $\mathbf{F}_{\mathcal{C}_j} \in \mathbb{R}^{H \times W}$ for the j$^{\text{th}}$ instance by computing $f_{\mathcal{C}_j}\left(\mathbf{e}_i\right)\ \forall\, i \in \{1, \ldots, |\mathcal{X}|\}$.

Intuitively, we want to obtain offset vectors for all pixels belonging to the instance $\mathcal{C}_j$ that point directly toward $\mathbf{c}_j$ while the remaining offset vectors should not. We achieve this implicitly by maximizing the intersection over union (IoU) between each annotated instance's soft and ground truth mask by minimizing the Lovász Hinge loss [19]:

$$\mathcal{L}_{\text{off}}\left(\mathbf{F}, \mathbf{G}, \mathcal{C}\right) = \frac{1}{|\mathcal{C}|} \sum_{j=1}^{|\mathcal{C}|} \text{Lovász}\left(\mathbf{F}_{\mathcal{C}_j}, \mathbf{G}_{\mathcal{C}_j}\right). \quad (3)$$

Here, $\mathbf{G}_{\mathcal{C}_j} \in \{0, 1\}^{H \times W}$ denotes the binary ground truth mask of the j$^{\text{th}}$ instance derived from its corresponding annotation $\mathcal{C}_j$. By maximizing the IoU, we implicitly encourage the network to predict for all pixels associated with a specific instance offset vectors that point toward their corresponding centroid. Simultaneously, this objective penalizes offset vectors that point toward a different centroid. Ultimately, the spatial embeddings of individual instances form clusters in 2D embedding space, see Fig. 2.

E. Postprocessing for Panoptic Segmentation

At inference time, the embeddings enable us to recover individual instances by applying an automated post-processing step. Initially, we compute the spatial embeddings for all pixels based on the predicted vector field. However, we ignore any pixel associated with the background based on the semantic segmentation. As a result, we obtain a 2D embedding space, where individual instances form separate clusters. Thus, we can employ DBSCAN [5] clustering to obtain an instance segmentation, see Fig. 2. Additionally, we propose a fusion step to unify the semantic and instance segmentation. Intuitively, each instance must belong to a unique semantic class. Thus, we gather all pixels in the semantic segmentation belonging to a specific instance and perform a majority voting to reassign pixels to the most frequent semantic class.

F. Training Dataset with Partial Annotations

In the partially supervised setting, we have access to a training dataset $\mathcal{D}^P = \left\{\left(\mathbf{X}, \mathbf{Y}^P, \mathcal{C}^P\right)\right\}$ containing the same images as $\mathcal{D}$, i.e., $|\mathcal{D}^P| = |\mathcal{D}|$. However, unlike before, we only have access to partial annotations $\mathbf{Y}^P$ and $\mathcal{C}^P$ for each image, as illustrated in Fig. 1. Let $\mathbf{Y}^P \in \mathbb{N}^{H\times W\times K}$ be the partial semantic annotation containing only for a subset $\mathcal{A} \subset \mathcal{X}$ of pixels the ground truth class, where $|\mathcal{A}| \ll |\mathcal{X}|$. Furthermore, we annotate only a subset of R instances $\mathcal{C}^P = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_R\}$ in each training image $\mathbf{X}$, where $|\mathcal{C}^P| \ll |\mathcal{C}|$. Here, we follow a particular annotation scheme to obtain these partial annotations. We instruct the labeler to annotate a certain percentage of all crop and weed instances in each image. Finally, the labeler uses a brush tool to sample a few regions in the background class. Thus, the partial annotation $\mathbf{Y}^P$ contains background regions, as shown in Fig. 1. Note that creating these partial annotations *for experimental evaluation* at varying annotation percentages is trivial using a densely annotated dataset.

G. Semantic Segmentation with Partial Annotations

Regardless of the annotation type, we employ our network's shared encoder and semantic-specific decoder to compute $\mathbf{S} = \phi_{\text{sem}}\left(\phi_{\text{enc}}\left(\mathbf{X}\right)\right)$, as described in Sec. III-C.

However, given $\mathcal{D}^P$, we adapt the cross-entropy loss in Eq. (1) to operate on partial annotations. Since this objective performs an independent comparison per pixel, we define the loss just over the subset $\mathcal{A}$ of annotated pixels, i.e., $\mathcal{L}_{\text{sem}}\left(\mathbf{Y}^P, \mathbf{S}, \mathcal{A}\right)$. Minimizing this objective still results in a reliable semantic segmentation [10].

H. Instance Segmentation with Partial Annotations

As before, despite the annotation type, we use the shared encoder and instance-specific decoder to compute the predicted vector field as $\mathbf{O} = \phi_{\text{off}}\left(\phi_{\text{enc}}\left(\mathbf{X}\right)\right)$.

Next, we adapt Eq. (3) to partially annotated images, containing a reduced set $|\mathcal{C}^P| \ll |\mathcal{C}|$ of annotated instances. Specifically, we define the loss function just for the set $\mathcal{C}^P$ of instances associated with an annotation, i.e., $\mathcal{L}_{\text{off}}\left(\mathbf{F}, \mathbf{G}, \mathcal{C}^P\right)$. By minimizing this objective, we implicitly encourage our network to predict offset vectors in $\mathbf{O}$ for all pixels associated with an annotated instance pointing toward their associated centroid. Furthermore, any offset vector, whether associated with an annotation or not, pointing toward an unrelated centroid of an annotated instance receives a penalty.

However, in our experiments, we observe that the model is not capable to generalize well from the set $\mathcal{C}^P$ of annotated instances to any instance $\mathcal{C}_i \in \mathcal{C} \setminus \mathcal{C}^P$. Generally, the offset vectors associated with $\mathcal{C}_i$ do not point toward a specific centroid, being inappropriate for clustering. This is a critical issue that we resolve by the method described subsequently.

I. Divergence for Vector Field Supervision

Since training with partial annotations does not yield an appropriate vector field, we need additional objectives to explicitly supervise the offset predictions of unannotated instances. To this end, we employ the *divergence* operator from vector calculus operating on a vector field and producing a scalar field, analyzing its behavior [16]. In 2D, the divergence $\operatorname{div} \mathbf{O} : \mathbb{R}^2 \to \mathbb{R}$ is defined as:

$$\operatorname{div} \mathbf{O} := \frac{\partial o_h\left(h, w\right)}{\partial h} + \frac{\partial o_w\left(h, w\right)}{\partial w}. \tag{4}$$

First, we provide an intuition about this operator and then explore its behavior for specific vector fields. Finally, we exploit it to define novel loss functions that encourage our network to predict offset vectors suitable for recovering all instances through clustering, whether annotated or not.

The divergence is particularly viscerally understood when imagining the vector field as a fluid flow, where fluid particles traverse along the offset vectors. Here, a positive divergence indicates a source from which all particles flow away. Contrary, a negative divergence implies a sink to which particles are flowing. This analogy transfers to our task since we want a vector field where offset vectors associated with a specific instance point toward this instance's centroid, i.e., a sink in the context of a fluid flow, as shown in Fig. 3. Thus, divergence is a useful tool for designing a loss function for this type of learning problem.

Let us analyze this operator in greater depth by investigating an illustrative and simplified vector field defined by a continuous multivariable function $\mathbf{O}\left(h, w\right) : \mathbb{R}^2 \to \mathbb{R}^2$ as:

$$\mathbf{O}\left(h, w\right) = \begin{bmatrix} o_h\left(h, w\right) \\ o_w\left(h, w\right) \end{bmatrix} = \begin{bmatrix} -h + c_h \\ -w + c_w \end{bmatrix}, \tag{5}$$

which behaves as a perfect sink, i.e., all offset vectors point toward $\mathbf{c} = [c_h, c_w]^\mathsf{T} \in \mathbb{R}^2$. Hence, it is evident that $\frac{\partial o_h(h,w)}{\partial h} = \frac{\partial o_w(h,w)}{\partial w} = -1$ and thus $\operatorname{div} \mathbf{O} = -2$.

Next, we expand our analysis to a discrete vector field that our network should predict for instance segmentation. Firstly, we generate an illustrative ground truth vector field containing offset vectors pointing directly toward their associated centroid, see Fig. 3. Secondly, we employ gradient filters to compute this vector field's partial derivatives and divergence. We highlight that the results are identical to the simplified example, as shown by the visualized partial derivatives in Fig. 3. However, we notice a few outliers in the transition area of overlapping instances. Here, adjacent offset vectors of different instances point toward different centroids, causing these outliers.

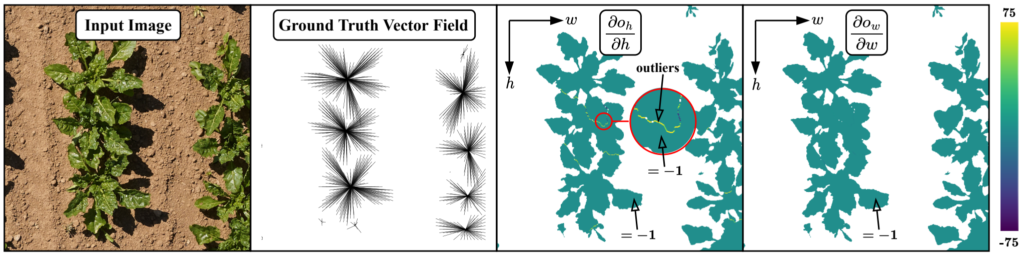

Fig. 3: We analyze the divergence of a vector field that our network should predict, i.e., each offset vector points toward the centroid of its associated plant instance. Specifically, we show that both partial derivatives are mostly equal to -1, except in transition areas of overlapping plants, where offset vectors point toward different centroids. During training, this enables us to define a set of loss functions applicable to instances without annotation, encouraging our network to predict the desired vector field even in the case of partial annotations.

The previous analysis provides desired properties a predicted vector field should have for our task. Thus, we define additional loss functions to encourage the network to predict a vector field with these properties. First, we define a robust regression loss function targeting the divergence of the predicted vector field during training:

$$\mathcal{L}_{\text{div}}\left(\mathbf{O}, \mathcal{V}\right) = \frac{1}{|\mathcal{V}|} \sum_{(h,w)\in\mathcal{V}} \rho\left(\left(\operatorname{div}\mathbf{O}\right)_{hw} - (-2)\right), \tag{6}$$

where $\mathcal{V} \subset \mathcal{X}$ is the set of pixel coordinates assigned to vegetation, encompassing both crops and weeds, as determined by the current semantic segmentation. Let $\rho\left(\cdot\right)$ denote the Geman-McClure loss [1] enforcing robustness to decrease the effect of expected outliers. Since this loss applies to any vegetation pixel, it affects the predicted offset vectors of any instance, whether annotated or not. Thus, it is well-suited in the case of partial annotations to also supervise the offset vectors of instances without annotation.

Simultaneously, we aim to predict identical partial derivatives, as shown previously. Consequently, we define a second robust regression loss function:

$$\mathcal{L}^{\text{aux}}_{\text{div}}(\mathbf{O}, \mathcal{V}) = \frac{1}{|\mathcal{V}|} \sum_{(h,w)\in\mathcal{V}} \rho\left(\frac{\partial o_h\left(h,w\right)}{\partial h} - \frac{\partial o_w\left(h,w\right)}{\partial w}\right), \tag{7}$$

encouraging the network to behave as desired.

The proposed formulation of both losses guarantees that an offset vector is only optimized if *both* partial derivatives are likely to be inliers. Conversely, the corresponding offset vector is not optimized if either is considered an outlier.

During training, we minimize both previous objectives to encourage our network to predict a vector field, yielding offset vectors applicable to recover instances through clustering.

J. Curl for Vector Field Supervision

We additionally employ the *curl* operator [16] to further supervise the offset predictions during training. As before, this operator generates a scalar field based on a vector field. In 2D, the operator $\operatorname{curl}\mathbf{O} : \mathbb{R}^2 \to \mathbb{R}$ is defined as:

$$\operatorname{curl}\mathbf{O} := \frac{\partial o_h\left(h,w\right)}{\partial w} - \frac{\partial o_w\left(h,w\right)}{\partial h}. \tag{8}$$

We again start by providing insight into the curl and study its behavior. Ultimately, we exploit this operator to define another set of loss functions to obtain a vector field with desirable properties for instance segmentation.

Generally, a positive curl indicates a rotational, counterclockwise behavior in a vector field, while a negative curl suggests a clockwise rotation. Apparently, we want to obtain a vector field with no rotational behavior for our task. This is evident when applying the curl to the illustrative vector field in Eq. (5) since $\frac{\partial o_h(h,w)}{\partial w} = \frac{\partial o_w(h,w)}{\partial h} = 0$ and $\operatorname{curl}\mathbf{O} = 0$.

As before, a thorough analysis based on a ground truth vector field similar to Fig. 3 exposes that the partial derivatives and curl are mostly the same as in the simplified example. Here, we again observe outliers in transition areas where offset vectors point toward different instances' centroids.

Once more, during training, we propose a robust regression loss function that focuses on the curl of the vector field:

$$\mathcal{L}_{\text{curl}}\left(\mathbf{O}, \mathcal{V}\right) = \frac{1}{|\mathcal{V}|} \sum_{(h,w)\in\mathcal{V}} \rho\left(\left(\operatorname{curl}\mathbf{O}\right)_{hw}\right), \tag{9}$$

applying to any predicted vegetation pixel.

Simultaneously, we aim to achieve identical partial derivatives and thus define an additional robust regression loss:

$$\mathcal{L}^{\text{aux}}_{\text{curl}}(\mathbf{O}, \mathcal{V}) = \frac{1}{|\mathcal{V}|} \sum_{(h,w)\in\mathcal{V}} \rho\left(\frac{\partial o_h\left(h,w\right)}{\partial w} + \frac{\partial o_w\left(h,w\right)}{\partial h}\right). \tag{10}$$

By minimizing both previous objectives during training, we encourage our network to predict a vector field with the desired behavior regarding its curl. Again, both losses are appropriate for partial annotations by supervising the offset vectors of any vegetation pixel, whether annotated or not.

K. Self-Consistency for Vector Field Supervision

Intuitively, the network should predict consistent vector fields for different content-preserving augmentations of the same input image. Thus, we propose a self-consistency loss, well-known to achieve smoother and more stable predictions. Additionally, this loss is well-suited to partial annotations since consistency does not rely on annotations.

To this end, we first apply a random augmentation to an input image to obtain $\mathbf{X}' = r_\theta\left(c\left(\mathbf{X}\right)\right)$ as its augmented view. Let $c\left(\cdot\right)$ denote a random color transformation and $r_\theta\left(\cdot\right)$ a random rotation, where $\theta = \{90°, 180°, 270°\}$. Thus, $\mathbf{X}'$ preserves the content of the original input, even though identical content appears at different locations.

Consequently, we pass both input images to our network to obtain $\mathbf{O} = \phi_{\text{off}}\left(\phi_{\text{enc}}\left(\mathbf{X}\right)\right)$ and $\mathbf{O}' = \phi_{\text{off}}\left(\phi_{\text{enc}}\left(\mathbf{X}'\right)\right)$ as their respective vector field predictions. Next, we compare both vector fields pixel-wise since they should be identical.

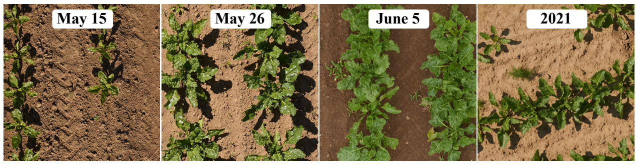

Fig. 4: Sample images from PhenoBench at different dates.

However, to enable a direct comparison, we first apply the inverse rotation $\mathbf{O}'' = r_\theta^{-1}(\mathbf{O}')$, ensuring alignment of offset vectors. Finally, we define a regression loss function encouraging the network to predict consistent vector fields:

$$\mathcal{L}_{\text{con}}(\mathbf{O}, \mathbf{O}'') = \frac{1}{|\mathcal{V}|} \sum_{(h,w)\in\mathcal{V}} ||\mathbf{O}_{hw} - \mathbf{O}''_{hw}||^2. \quad (11)$$

L. Implementation Details

We use the ERFNet [23] architecture and convert it into a 2-branch network, where the first branch predicts the semantic segmentation and the second the vector field. During training, we employ the Adam optimizer [11] with a batch size of 6 and a weight decay of 10^{-4}. Here, we train the model for 4096 epochs. At the initial 32 epochs we linearly increase the learning rate from 0 to 10^{-3} and then apply a polynomial learning rate decay $\left(1 - \frac{e}{4096}\right)^2$, where e is the current epoch. We define the final loss as a weighted sum:

$$\mathcal{L} = \mathcal{L}_{\text{sem}} + w_1 \mathcal{L}_{\text{off}} + w_2 \left(\bar{\mathcal{L}}_{\text{div}} + \bar{\mathcal{L}}_{\text{curl}} + w_3 \mathcal{L}_{\text{con}}\right), \quad (12)$$

where $\bar{\mathcal{L}}_{\text{div}} = \mathcal{L}_{\text{div}} + \mathcal{L}_{\text{div}}^{\text{aux}}$ and $\bar{\mathcal{L}}_{\text{curl}} = \mathcal{L}_{\text{curl}} + \mathcal{L}_{\text{curl}}^{\text{aux}}$. Let w_1 and w_3 denote constant weights set to 5 and 10, respectively. In contrast, we quadratically increase w_2 from 0 to 1 during the initial 128 epochs and set it to 1 afterward. We determine these hyperparameters based on the validation set.

IV. Experimental Evaluation

Following, we provide a comprehensive experimental evaluation, supporting our key claims made in Sec. I.

Datasets. We evaluate our method on the PhenoBench dataset [27] containing real-world RGB images captured from a nadir-view perspective of sugar beet fields and providing dense annotations for semantic and instance segmentation of crops and weeds. The images contain plants at different growth stages throughout the year 2020 from May 15, May 26, and June 6, as shown in Fig. 4. Hence, the overlap between adjacent plant instances increases over time. Besides, the dataset contains test images from a sugar beet field captured in 2021 to evaluate generalization capabilities.

First, we automatically generate partial annotations using the original dense annotations simulating a human labeler, as proposed in Sec. III-F. Besides, this dataset provides the visibility ratio for each instance. We exploit this to leave any instance with a visibility ratio of $< 25\,\%$ unannotated, as it would pose a challenge for a human labeler to assign such an instance to either crop or weed. We denote a partially annotated dataset $\mathcal{D}^P$, with $50\,\%$, $25\,\%$, or $10\,\%$ of all crop and weed instances annotated as $\mathcal{D}^{50}$, $\mathcal{D}^{25}$, or $\mathcal{D}^{10}$.

Note that consecutive images in the original dataset overlap by $50\,\%$, resulting in single plant instances appearing in multiple images. However, in our partial annotation setup, we aim to ensure that an instance without annotation in one image remains unannotated in any other image. Thus, we remove overlapping images in the training split and obtain a reduced set of 356 images for training. However, we use the original validation and test set for our experiments. For more information about the dataset statistics, we refer to [27].

Evaluation Metrics. To evaluate the performance for panoptic segmentation we follow the PhenoBench setup [27] and compute the panoptic quality [12] for crop and weed instances, denoted as PQ_c and PQ_w, respectively. Furthermore, we compute the intersection over union for the background class, i.e., IoU_bg. Finally, we report $\text{PQ}^\dagger$ as average over all metrics [27]. We also compute the mean intersection over union (mIoU) over background, weed, and crop to evaluate the semantic segmentation. Note that for all metrics higher values indicate better performance.

A. Performance based on Partial Annotations

In the first experiment, we show that our proposed method achieves high performance when trained using partial annotations, thus supporting our first key claim. To this end, we train a model based on dataset $\mathcal{D}$ with dense annotations and compare its performance with models trained on $\mathcal{D}^{50}$, $\mathcal{D}^{25}$, and $\mathcal{D}^{10}$ containing partial annotations. We employ all previously introduced loss functions during the training of each model. Thus, the only difference are reduced annotations.

In Tab. I, we show that our model can attain high performance even with substantially fewer annotations during training. Specifically, concerning May 15, the model's performance trained on $\mathcal{D}$ is similar to that on $\mathcal{D}^{25}$ achieving a $\text{PQ}^\dagger$ score of $70.48\,\%$ and $70.06\,\%$, respectively. Thus, our approach substantially reduces the required annotations at this early growth stage, where individual plants are mostly well separated. Contrary, the performance gap increases at later growth stages, as expected. Here adjacent plants overlap considerably. Particularly, regarding June 6, we achieve a $\text{PQ}^\dagger$ score of $77.24\,\%$ when employing $\mathcal{D}$ for training and $73.53\,\%$ when reyling on $\mathcal{D}^{10}$. However, we highlight a marginal performance drop in these challenging conditions between the models trained on $\mathcal{D}^{50}$ and $\mathcal{D}^{10}$ differing by just 1.03 absolute percent points. Furthermore, we highlight that specifically on June 6, small and large plants co-occur in the same image [27], but our model still performs reliably, as evident in Tab. I. Additionally, when evaluated in another field in 2021, the model trained on $\mathcal{D}^{50}$ achieves a competitive performance regarding its generalization capabilities compared with the model trained on $\mathcal{D}$, achieving a $\text{PQ}^\dagger$ score of $74.44\,\%$ and $74.75\,\%$ each.

B. Ablation Study on Vector Field Supervision

Next, we show that our proposed losses to supervise the predicted vector field during training effectively enhance the performance of our model, supporting our second claim. Specifically, we evaluate the influence of the different losses on the validation set of PhenoBench, exclusively consisting of images captured in the year 2020, with none from 2021.

TABLE I: Evaluation of varying partial annotations on the test set of PhenoBench. We report all metrics in percentage.

	Dataset	$PQ^{\dagger}$	PQ_c	PQ_w	IoU_{bg}	mIoU
May 15	$\mathcal{D}$	**70.48**	74.76	**36.99**	**99.68**	**81.48**
	$\mathcal{D}^{50}$	**70.48**	**75.20**	36.57	**99.68**	81.36
	$\mathcal{D}^{25}$	70.06	73.97	36.55	99.67	80.46
	$\mathcal{D}^{10}$	68.91	72.97	34.13	99.64	79.33
May 26	$\mathcal{D}$	**77.18**	**84.17**	**48.51**	**98.87**	**86.21**
	$\mathcal{D}^{50}$	76.16	82.94	46.74	98.80	85.30
	$\mathcal{D}^{25}$	75.09	80.66	45.92	98.70	84.31
	$\mathcal{D}^{10}$	73.69	78.43	44.20	98.43	82.94
June 6	$\mathcal{D}$	**77.24**	**83.84**	**49.32**	**98.56**	**90.96**
	$\mathcal{D}^{50}$	74.56	78.75	46.63	98.30	88.99
	$\mathcal{D}^{25}$	74.23	77.30	47.31	98.09	83.77
	$\mathcal{D}^{10}$	73.53	76.88	45.86	97.84	82.51
2021	$\mathcal{D}$	**74.75**	**86.20**	40.08	**97.96**	78.96
	$\mathcal{D}^{50}$	74.44	83.36	**42.05**	97.92	**79.47**
	$\mathcal{D}^{25}$	71.07	79.87	35.53	97.82	76.18
	$\mathcal{D}^{10}$	67.68	71.71	33.72	97.61	75.97

First, we train a model based on $\mathcal{D}^{25}$ without additional vector field supervision. Following, we compare its performance to models trained with increasing supervision by successively including the losses targeting the divergence, curl, and consistency. Finally, we leverage our entire pipeline and compare to a model trained with all objectives and also include our fusion procedure, as explained in Sec. III-E.

In Tab. II, we show that each of our proposed losses to supervise the predicted vector field increases the performance when training with partial annotations. Note that the additional supervision is particularly beneficial in challenging conditions with overlapping plants, e.g., on June 6. Particularly, without extra supervision we achieve a $PQ^{\dagger}$ score of $69.18\,\%$ while the model trained with all proposed losses achieves $75.40\,\%$. We support this quantitative evaluation with qualitative results in Fig. 5, showcasing how our proposed losses effectively supervise the offset vectors towards the desired behavior, resulting in spatial embeddings that are well-suited for clustering. However, the mIoU decreases slightly from $86.53\,\%$ to $86.15\,\%$. In less complex situations, as on May 15, we observe a less drastic improvement regarding the $PQ^{\dagger}$ score from $67.07\,\%$ to $69.05\,\%$. We highlight that our proposed fusion procedure in Sec. III-E improves the performance further across all dates.

C. Comparison to Baselines

To support our third claim, we show that our approach outperforms related methods targeting to reduce the annotation bottleneck. Specifically, we compare with weakly supervised methods, i.e., DiscoBox [14] and BoxInst [25]. Additionally, we train a state-of-the-art object detection model based on DETR [2] and pass its predicted bounding boxes as prompts to the foundation model SAM [13], providing an instance segmentation for each prompt. Lastly, we compare against Mask R-CNN [7]. We train this model using the partial annotations of $\mathcal{D}^{10}$ by explicitly ignoring instances without annotations to prevent any penalties during training.

In Tab. III, we show that the combination DETR $\rightarrow$ SAM performs most competitively compared to our approach. Particularly, both methods achieve a $PQ^{\dagger}$ score of $73.73\,\%$

TABLE II: Ablation study of increasing vector field supervision and our proposed fusion procedure based on models trained with $\mathcal{D}^{25}$. We report all results on the PhenoBench validation set and report each metric in percentage.

	Div	Curl	Con	Fuse	$PQ^{\dagger}$	PQ_c	PQ_w	IoU_{bg}	mIoU
May 15					67.07	70.50	31.06	99.64	79.51
	✓				67.83	71.16	32.69	99.63	79.33
	✓	✓			68.05	70.92	33.62	99.62	78.87
	✓	✓	✓		69.05	72.84	34.66	99.64	80.56
	✓	✓	✓	✓	**70.58**	**75.31**	**36.76**	**99.68**	**81.10**
May 26					69.99	72.51	38.37	99.10	80.81
	✓				72.09	75.71	41.53	99.03	80.21
	✓	✓			71.84	76.16	40.35	99.02	80.65
	✓	✓	✓		73.33	78.25	42.65	99.10	81.50
	✓	✓	✓	✓	**75.31**	**81.16**	**45.54**	**99.22**	**82.00**
June 6					69.18	61.98	47.51	98.04	86.53
	✓				71.83	71.45	46.36	97.68	84.80
	✓	✓			72.93	74.71	46.41	97.67	83.86
	✓	✓	✓		75.40	76.77	51.61	97.82	86.15
	✓	✓	✓	✓	**79.24**	**82.14**	**57.42**	**98.16**	**86.66**

and $73.53\,\%$ when evaluated on June 6, respectively. However, the performance gap increases in earlier growth stages, where plants have finer structures, e.g., on May 15. The former method achieves a $PQ^{\dagger}$ score of $64.64\,\%$, whereas our approach achieves $68.91\,\%$.

D. Performance on Other Datasets

The last experiment shows the transferability of our method to different datasets. We evaluate our model based on the popular CVPPP leaf segmentation challenge (LSC) [24] containing images captured from a nadir-view perspective in a laboratory environment. Here, we compare our results with another competitive baseline, i.e., SPOCO [28].

In Tab. IV, we show that both methods achieve comparable performance on the commonly used evaluation metric when trained with dense annotations or the most reduced set of annotations, i.e., $\mathcal{D}^{10}$. Contrary, our method substantially outperforms the baseline in the case of models trained on $\mathcal{D}^{25}$, achieving a score of $86.13\,\%$ and $83.50\,\%$, respectively. We observe similar behavior for models trained on $\mathcal{D}^{50}$, suggesting that our approach handles these partial annotations more effectively. Note that SPOCO performs a class-agnostic instance segmentation but is not designed to differentiate different classes, e.g., crops and weeds.

V. Conclusion

In this paper, we present a novel vision-based method targeting panoptic segmentation in the agricultural domain that substantially reduces the annotation bottleneck. We propose a CNN that performs a semantic segmentation and jointly predicts a vector field where offset vectors of individual plant instances point toward their associated centroid. This enables us to perform an instance segmentation through clustering. Our approach is trainable on partial annotations, where only a subset of pixels is annotated. To this end, we propose a novel set of losses based on common operators used in physics to analyze the behavior of vector fields, i.e., divergence and curl. We exploit these operators to effectively supervise predicted offsets associated with no annotation to obtain high performance even with substantially reduced annotations.

TABLE III: Evaluation on the test set of PhenoBench with various approaches to reduce the annotations; all metrics in percentage.

	Approach	$PQ^{\dagger}$	PQ_c	PQ_w	IoU_{bg}	mIoU
May 15	BoxInst	57.41	54.26	18.50	99.46	75.62
	DiscoBox	48.55	36.17	10.54	98.94	70.14
	DETR → SAM	64.64	62.27	32.10	99.56	79.10
	Mask R-CNN	58.13	57.08	17.91	99.40	69.53
	Ours $(\mathcal{D}^{10})$	**68.91**	**72.97**	**34.13**	**99.64**	**79.33**
May 26	BoxInst	67.84	74.72	30.89	97.92	78.85
	DiscoBox	57.50	55.44	20.92	96.15	74.95
	DETR → SAM	68.19	74.09	32.36	98.11	80.70
	Mask R-CNN	65.88	67.14	33.73	96.76	74.78
	Ours $(\mathcal{D}^{10})$	**73.69**	**78.43**	**44.20**	**98.43**	**82.94**
June 6	BoxInst	70.07	74.30	38.99	96.91	84.61
	DiscoBox	58.62	55.06	26.53	94.26	74.42
	DETR → SAM	**73.73**	74.88	**49.20**	97.11	**86.21**
	Mask R-CNN	69.23	71.17	42.21	94.32	73.10
	Ours $(\mathcal{D}^{10})$	73.53	**76.88**	45.86	**97.84**	82.51
2021	BoxInst	59.47	50.49	33.15	94.76	70.60
	DiscoBox	45.63	23.92	20.50	92.46	66.44
	DETR → SAM	66.35	58.18	**44.43**	96.45	**78.53**
	Mask R-CNN	52.23	39.14	23.54	94.02	64.61
	Ours $(\mathcal{D}^{10})$	**67.68**	**71.71**	33.72	**97.61**	75.97

TABLE IV: Evaluation of varying methods on the test set of CVPPP LSC with partial annotations using Symmetric Best Dice (SBD).

Approach	Dataset	SBD [%]
SPOCO / Ours	$\mathcal{D}$	**87.17** / 86.25
SPOCO / Ours	$\mathcal{D}^{50}$	83.50 / **86.13**
SPOCO / Ours	$\mathcal{D}^{25}$	81.30 / **84.66**
SPOCO / Ours	$\mathcal{D}^{10}$	**77.55** / 77.23

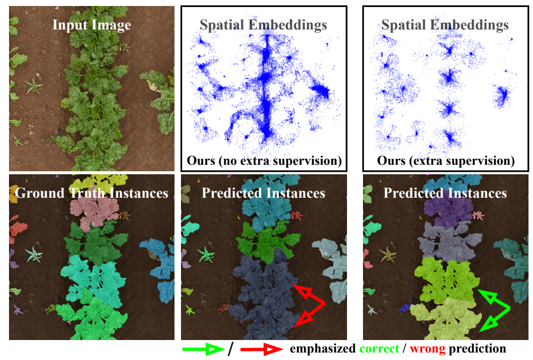

Fig. 5: Qualitative results on PhenoBench. Left: Input Image and annotated instances. Middle: Spatial embeddings of our model $(\mathcal{D}^{25})$ without extra vector field supervision and predicted instances after clustering. Right: Same as before but with our proposed supervision.

References

[1] J.T. Barron. A General and Adaptive Robust Loss Function. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

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

[4] B. Cheng, M. Collins, Y. Zhu, T. Liu, T. Huang, H. Adam, and L. Chen. Panoptic-DeepLab. *arXiv preprint*, arXiv:1910.04751, 2019.

[5] M. Ester, H. Kriegel, J. Sander, and X.Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the Conf. on Knowledge Discovery and Data Mining (KDD)*, 1996.

[6] M. Halstead, A. Ahmadi, C. Smitt, O. Schmittmann, and C. McCool. Crop agnostic monitoring driven by deep learning. *Frontiers in Plant Science*, 12, 2021.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2017.

[8] C.C. Hsu, K.J. Hsu, C.C. Tsai, Y.Y. Lin, and Y.Y. Chuang. Weakly supervised instance segmentation using the bounding box tightness prior. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2019.

[9] W. Ji, J. Li, Q. Bi, W. Li, and L. Cheng. Segment Anything is Not Always Perfect: An Investigation of SAM on Different Real-World Applications. *arXiv preprint arXiv:2304.05750*, 2023.

[10] D. Kigli, A. Shamir, and S. Avidan. How Low Can We Go? Pixel Annotation for Semantic Segmentation. *arXiv preprint arXiv:2201.10448*, 2022.

[11] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2015.

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

[13] 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 *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2023.

[14] S. Lan, Z. Yu, C. Choy, S. Radhakrishnan, G. Liu, Y. Zhu, L.S. Davis, and A. Anandkumar. DiscoBox: Weakly Supervised Instance Segmentation and Semantic Correspondence from Box Supervision. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[15] P. Lottes, J. Behley, A. Milioto, and C. Stachniss. Fully Convolutional Networks with Sequential Information for Robust Crop and Weed Detection in Precision Farming. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):3097–3104, 2018.

[16] P.C. Matthews. *Vector Calculus*. Springer Science & Business Media, 1 edition, 2012.

[17] C. McCool, T. Perez, and B. Upcroft. Mixtures of Lightweight Deep Convolutional Neural Networks: Applied to Agricultural Robotics. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[18] A. Milioto, P. Lottes, and C. Stachniss. Real-time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[19] D. Neven, B.D. Brabandere, M. Proesmans, and L.V. Gool. Instance Segmentation by Jointly Optimizing Spatial Embeddings and Clustering Bandwidth. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[20] K.D. Nguyen, T.H. Phung, and H.G. Cao. A SAM-based Solution for Hierarchical Panoptic Segmentation of Crops and Weeds Competition. In *Proc. of the Int. Conf. on Computer Vision Workshops*, 2023.

[21] D. Petti and C. Li. Weakly-supervised Learning to Automatically Count Cotton Flowers from Aerial Imagery. *Computers and Electronics in Agriculture*, 194:106734, 2022.

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

[23] E. Romera, J.M. Alvarez, L.M. Bergasa, and R. Arroyo. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation. *IEEE Trans. on Intelligent Transportation Systems (T-ITS)*, 19(1):263–272, 2018.

[24] H. Scharr, M. Minervini, A. Fischbach, and S.A. Tsaftaris. Annotated Image Datasets of Rosette Plants. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2014.

[25] Z. Tian, C. Shen, X. Wang, and H. Chen. BoxInst: High-Performance Instance Segmentation with Box Annotations. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[26] J. Weyler, T. Läbe, F. Magistri, J. Behley, and C. Stachniss. Towards Domain Generalization in Crop and Weed Segmentation for Precision Farming Robots. *IEEE Robotics and Automation Letters (RA-L)*, 8(6):3310–3317, 2023.

[27] 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. *arXiv preprint arXiv:2306.04557*, 2023.

[28] A. Wolny, Q. Yu, C. Pape, and A. Kreshuk. Sparse Object-Level Supervision for Instance Segmentation With Pixel Embeddings. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

CERTIFICATE OF REPRODUCIBILITY

The authors of this publication declare that:

1) The software related to this publication is distributed in the hope that it will be useful, support open research, and simplify the reproducability of the results but it comes without any warrenty and without even the implied warranty of merchantability or fitness for a particular purpose.

2) *Jan Weyler* primarily developed the implementation related to this paper. This was done on Ubuntu 22.04.

3) *Jens Behley* verified that the code can be executed on a machine that follows the software specification given in the Git repository available at:

   https://github.com/PRBonn/PSPA

4) *Thomas Läbe* verified that the experimental results presented in this publication can be reproduced using the implementation used at submission, which is labeled with a tag in the Git repository and can be retrieved using the command:

   ```
   git checkout RAL
   ```