# Few-Shot Adaptation of Grounding DINO for Agricultural Domain

Rajhans Singh, Rafael Bidese Puhl, Kshitiz Dhakal, Sudhir Sornapudi
Corteva Agriscience, Indianapolis, USA
{rajhans.singh, rafael.bidesepuhl, kshitiz.dhakal-1, sudhir.sornapudi}@corteva.com

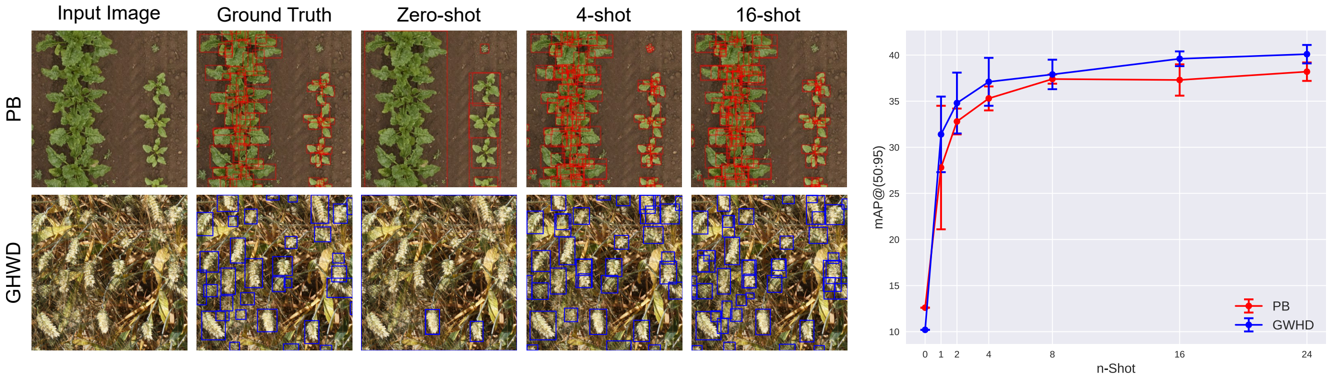


Figure 1. Left: zero-shot vs our few-shot (4-shot, 16-shot) using Grounding DINO on Wheat Head (GWHD) [12] and PhenoBench (PB) [62]. Zero-shot fails in cluttered/occluded environments, whereas our few-shot outperforms significantly. Right: mAP of our few-shot approach increases with more training images.

## Abstract

*Deep learning models are transforming agricultural applications by enabling automated phenotyping, monitoring, and yield estimation. However, their effectiveness heavily depends on large amounts of annotated training data, which can be labor and time intensive. Recent advances in open-set object detection, particularly with models like Grounding-DINO, offer a potential solution to detect regions of interests based on text prompt input. Initial zero-shot experiments revealed challenges in crafting effective text prompts, especially for complex objects like individual leaves and visually similar classes. To address these limitations, we propose an efficient few-shot adaptation method that simplifies the Grounding-DINO architecture by removing the text encoder module (BERT) and introducing a randomly initialized trainable text embedding. This method achieves superior performance across multiple agricultural datasets, including plant-weed detection, plant counting, insect identification, fruit counting, and remote sensing tasks. Specifically, it demonstrates up to a* $\sim 24\%$ *higher mAP than fully fine-tuned YOLO models on agricultural datasets and outperforms previous state-of-the-art methods by* $\sim 10\%$ *in remote sensing, under few-shot learning conditions. Our method offers a promising solution for automating annotation and accelerating the development of specialized agricultural AI solutions.*

## 1. Introduction

Deep learning based object detection [7, 41, 42] and segmentation [9, 19] models are increasingly important in agriculture for tasks like phenotyping [58, 61, 62], monitoring [4, 6], pest detection [1, 8], yield estimation [11, 24], etc. Due to the vast diversity of the agricultural domain, unique tasks and imaging setups, these models need dedicated, extensive and varied training data to ensure reliable performance. However, manual annotation of such datasets is time-consuming and costly, creating a major bottleneck in developing effective AI solutions. Streamlining or automating the annotation process would significantly improve the efficiency and scalability of these models, benefiting the agricultural sector.

Recent advances [10, 15, 16, 33, 51, 63] have focused on using large vision-language models for open-set object detection in computer vision. These models are designed to detect objects beyond a predefined set by using human language input, making them highly versatile. Notable among these is Grounding-DINO[33], which has been trained on very large datasets including O365[50], OI[25], GoldG[21], Cap4M[28], COCO[30], and RefC[23]. Building on this foundation, recent work [44] incorporated SAM2[39] to add segmentation capabilities, allowing the model to generate masks for detected objects.

A significant advantage of these models is their generalization ability, enabling them to detect unseen objects and adapt across various domains. Their large size, high computational cost, and slow inference speed pose challenges for real-time applications in agriculture. Despite these limitations, these models hold potential for automating annotation processes and distilling their knowledge into smaller, specialized models tailored for specific agricultural tasks.

In this paper, we explore the application of pretrained Grounding-DINO on wide-range of publicly available agriculture datasets, including plant-weed detection [2, 18], insect identification [54], wheat head detection [12], fruit counting [47, 55], and remote sensing [27]. We further investigate instance segmentation using Grounding-DINO combined with SAM2 on datasets such as PhenoBench[62].

First, we leverage zero-shot learning, where the model relies solely on text prompts corresponding to different classes without using any training images from the target datasets. However, we encounter several challenges: finding appropriate text prompts that consistently yield strong performance is difficult, particularly for objects like individual leaves in a plant due to overlapping structures. Additionally, distinguishing between visually similar classes, such as certain weeds resembling crop, proved challenging when generating text prompts. Finally, combining diverse-looking instances of the same class within a single text prompt adds further complexities.

To address the challenges associated with text prompting, we propose an efficient few-shot adaptation approach that enhances the performance of the Grounding-DINO model on agricultural datasets. Our method simplifies the original architecture by removing its language processing component (BERT) and introducing randomly initialized trainable parameters that mimic the shape of the BERT's text embeddings. This modification eliminates the need for text prompts, allowing us to adapt the model to specific datasets with minimal effort.

By learning these new text embeddings (a few thousand trainable parameters) with as few as two labeled images and iterating for a small number of training steps, we achieve excellent results across diverse agricultural datasets. This approach not only reduces the complexity of the model but also significantly speeds up the adaptation process for new datasets. We demonstrate superior performance compared to the zero-shot and other state-of-the-art few-shot approaches. This makes it particularly valuable in automating annotation tasks within agriculture and other fields where efficiency is critical.

Our main contributions can be summarized as follows:

- We investigate the application of Grounding-DINO model across diverse agricultural datasets for object detection task, leveraging its zero-shot capabilities to detect objects without requiring any labeled training data.
- We discover manual prompt tuning for agriculture applications is impractical due to significant challenges in crafting effective prompts for zero-shot approach in various scenarios.
- We propose a simple and efficient few-shot learning method for Grounding-DINO that eliminates the need for text prompts and enables efficient adaptation to new datasets using only a minimal number of training examples and iterations.
- We conduct a series of experiments across various agricultural datasets, demonstrating our few-shot approach's superior performance compared to zero-shot and other state-of-the-art few-shot methods.

## 2. Related Work

**Object detection in Agriculture.** Object detection plays a pivotal role in agriculture for tasks [5] including disease detection [66], crop identification [60], pest detection [57], fruit counting [46], and cattle tracking [53]. Current approaches often rely on supervised learning frameworks, fine-tuning benchmark models like YOLO variants [6] for real-time performance. However, these methods demand extensive annotated datasets, increasing costs and time requirements. They also exhibit limited generalization to novel classes due to their closed-set detection focus [14].

**Open-set Object Detection.** Open-set object detection extends traditional methods like Faster R-CNN [43] and YOLO [40] to address novel object detection, building on Scheirer et al.'s [48] open-set recognition framework. Recent advancements include Grounding-DINO [33], which leverages the DINO transformer and grounded pre-training for text-visual alignment, and YOLO-UniOW [32], demonstrating iterative vocabulary expansion to improve detection robustness. However, these models often struggle with agricultural datasets. To address this challenge, we opted for an efficient few-shot learning approach.

**Few-shot Object Detection.** Few-shot learning for object detection [17, 64] aims to detect and classify objects in images with only a few annotated examples per class, addressing the challenge of misalignment with target region of interest [36] and data scarcity [3]. This is reinforced by the boost in detection performance on small agricultural datasets [38]. Our paper demonstrates the benefits of leveraging few-shot learning over a zero-shot foundational detection model like Grounding-DINO.

**Prompt Learning.** Text input (prompt) plays an important role in vision-language models, but finding the right prompt can be a challenging process. To address this challenge, recent works [26, 52] have proposed prompt learning techniques to systematically improve prompt design. Prompt tuning has been mainly explored with CLIP models [37] for image classification [22, 67, 68]. For classification problems, given the template 'a photo of {label},' these methods utilize backpropagation to optimize the {label} token to match text and image features. Recent work by Li et al. [28] employs prompt learning for object detection using the GLIP model, where text prompts (e.g., 'detect fish') are utilized with the language encoder to generate text embeddings, which are then optimized using a few training images. Although, our approach is similar in concept, it differs in three key aspects: (1) we completely eliminate the use of text prompts for initialization and remove the text encoder (BERT) entirely; (2) our method initializes embeddings randomly and experiments with varying numbers of embeddings per class, demonstrating that performance improves as the number of embeddings per class increases; and (3) we apply this approach to pretrained Grounding-DINO model and benchmark it on agriculture-related datasets.

## 3. Method

In this section, we present our few-shot adaptation approach using Grounding-DINO for agricultural datasets. Our method leverages a pre-trained Grounding-DINO model and eliminates the need for text prompts, enabling direct optimization within its text feature space. This modification allows for efficient adaptation to various datasets with minimal effort. We provide detailed insights into our approach in the following subsections.

### 3.1. Grounding-DINO

Grounding-DINO is a popular large vision-language model equipped with open-set object detection capabilities. This enables the recognition of novel objects outside the initially defined categories.

As shown in Figure 2, the Grounding-DINO model processes an image along with a text prompt corresponding to each class. The model comprises several key components: an image backbone that extracts image features, a text backbone that encodes textual information into feature vectors, a feature enhancer module that fuses these image and text features, a language-guided query selection module that initializes object queries based on the input text prompt, and finally, a cross-modality decoder that refines object features and bounding boxes.

For each (image, text) pair, the model processes images using the Swin Transformer[35] and text via a BERT encoder [13]. Image features are hierarchically extracted from different layers of the Swin Transformer, capturing hierarchical visual information. Text is tokenized using the byte-pair encoding (BPE) scheme [49], which is then encoded by the BERT model to produce $N \times 768$-dimensional text embeddings (features), where $N$ represents the number of tokens in the text prompt. These raw image and text embeddings are fed into the feature enhancer module to enable cross-modal fusion. The module comprises multiple enhancer layers, each containing self-attention mechanisms for both image and text processing, as well as cross-attention layers that facilitate interactions between text-to-image and image-to-text contexts.

Grounding-DINO method uses the enhanced text features to select object queries by calculating the dot product between text and image features. It then selects $N_I$ object queries by choosing image features with the maximum scores. These language-guided queries are subsequently fed into a cross-modality decoder. In this decoder, query features undergo a series of self-attention operations, followed by cross-attentions with both image and text features. Finally, the output queries from the decoder's last layer are utilized to predict object boxes and corresponding class probabilities based on similarity with the text features.

For training, the model uses a contrastive loss between predicted objects and language token features. To compute this loss, each query calculates a dot product with text features to produce logits for every text token. Then the focal loss [31] is applied to these logits to obtain the classification loss. Additionally, L1 and GIoU[45] losses are employed for the bounding box regression. Similar to DETR-like models [7], bipartite matching is used to find the matching between predicted and ground-truth objects. The total loss is then computed using the combination of classification and bounding-box losses based on this mapping.

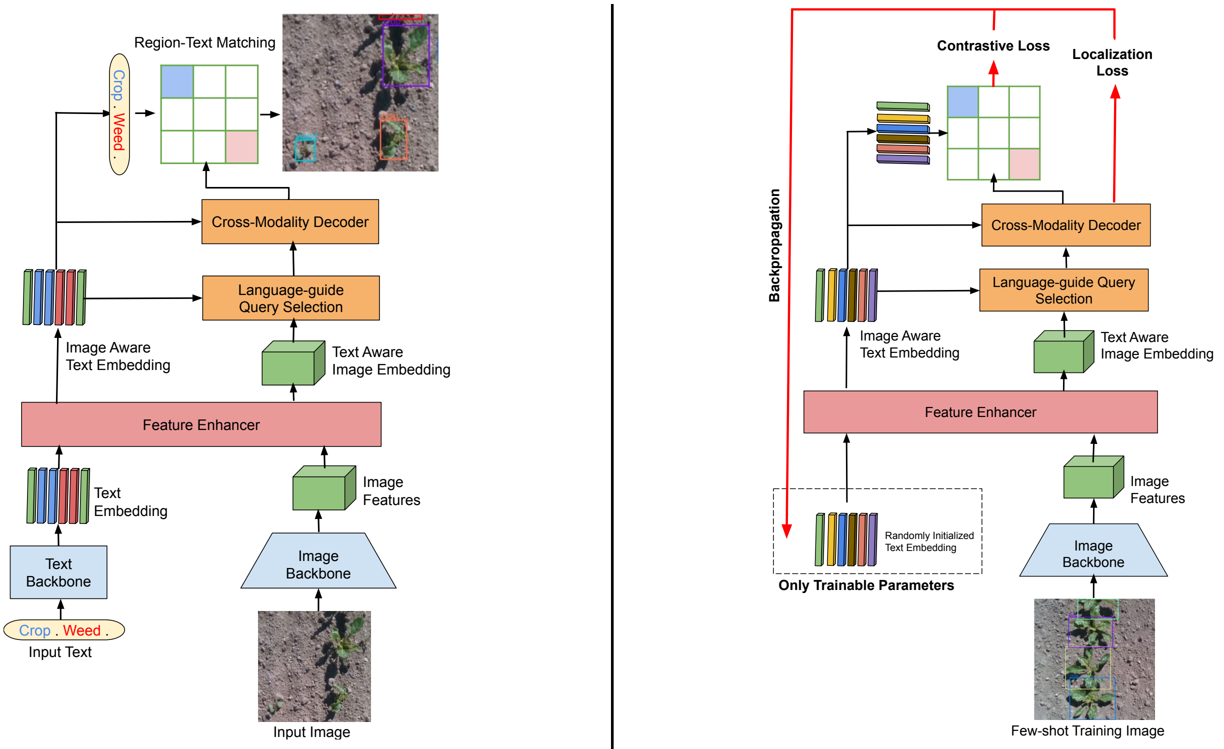
### 3.2. Zero-shot approach

In the zero-shot setting with Grounding-DINO, the pretrained model is evaluated on test sets from various datasets without fine-tuning. To construct text prompts for different classes, two approaches are employed: using single words separated by full stops (e.g., "crop . weed .") and creating phrases for each class separated by full stops (e.g., "green pepper . red pepper .") to enhance the distinction between similar categories. During class prediction, the model computes a dot product between text token features and predicted object features, identifying the highest score index. If this index corresponds to tokens within the class's text prompt set, the associated class is assigned.

### 3.3. Few-shot approach

In agricultural datasets, creating effective text prompts is particularly challenging due to their vastness and diversity, as well as the domain-specific nuances. To tackle this prompting challenge, we introduce a few-shot learning approach, where a small number of labeled images are used to adapt the pretrained grounding model for new datasets. This method is especially beneficial when labeled data is limited and enables a single pre-trained model to effectively handle diverse agricultural datasets by leveraging minimal examples for adaptation.

As shown in Figure 2, our approach eliminates the BERT-based text encoder (dashed box in the right panel) and operates directly within the output space of the BERT



Grounding-DINO

Proposed Few-shot Approach

Figure 2. Figure shows the block diagram of zero-shot inference with Grounding-DINO (left) and our proposed few-shot approach (right). Our method eliminates the BERT text encoder and operates directly in BERT's output space. Text embeddings are initialized randomly with dimensions matching BERT's outputs. We train only these text embeddings (approximately a few thousand parameters), keeping the rest of Grounding-DINO's parameters frozen, which requires a few labeled images and training iterations to achieve strong performance across diverse agricultural datasets.

model. Text embeddings are initialized randomly with dimensions matching BERT outputs, and position IDs along with attention masks are designed to ensure class-specific feature attention, mirroring BERT's mechanisms.

Similar to text prompts, a single class can have multiple words or tokens; in our setup, we use multiple embeddings per class. To maintain same shape as typical BERT model outputs, we include dummy embeddings for start and end tokens. Given C classes and T embeddings per class, the dimensionality of these text embeddings is $(C * T + 2) \times 768$, where $+2$ accounts for the start-end tokens and 768 represents the output dimension of the BERT model.

In our few-shot implementation, we fine-tune only the text embeddings while keeping all other parameters of the Grounding-DINO frozen. For a labeled image, the model outputs both class probabilities and bounding box coordinates. Let $N_I$ denote the number of object queries with feature vector $X_I$, and let $N_T$ represent the number of text embeddings with feature vector $X_T$. The output class probability matrix is computed as

$$P_{out} = \text{sigmoid}(X_I X_T^T)$$

where $P_{out}$ has dimensions $N_I \times N_T$. For ground-truth probabilities $P_{gt}$, we assign a value of 1 for all token indices corresponding to the correct class and 0 otherwise. We use focal loss [31] for classification, denoted by $L_{cls}$. For bounding box loss, we use $L1$ and GIoU loss [45], denoted by $L_{giou}$. Our total loss is given by

$$L = \lambda_1 L_{cls} + \lambda_2 L1 + \lambda_3 L_{giou}$$

where $\lambda_1$, $\lambda_2$, and $\lambda_3$ are hyper-parameters, empirically set to 1, 5, and 2, respectively. To match predicted objects $y$ with ground-truth objects $\hat{y}$, we use bipartite matching [7]. The optimal assignment $\hat{\sigma}$ is determined by minimizing

$$\hat{\sigma} = \arg\min_{\sigma} \sum_{i=1}^{N} L(y_i, \hat{y}_{\sigma(i)})$$

Based on this optimal matching, we compute the final loss and update the trainable text embedding parameter $\mathcal{W}$ using backpropagation

$$\mathcal{W}_{t+1} = \mathcal{W}_t - \eta \nabla_{\mathcal{W}} L_{\hat{\sigma}}(y, \hat{y})$$

where $\eta$ is the learning rate and $\nabla_{\mathcal{W}}$ represents the gradient of the loss function with respect to $\mathcal{W}$. After optimizing the text embedding parameters using a few training images for a few training iterations, $t$, we obtain an optimally tuned embedding vectors. These learned embeddings are used for inference on test data.

For text feature initialization, we employ a normal distribution. While these features can be initialized using text prompts input through the BERT model, our experiments show that random initialization achieves comparable performance. Unlike text prompt initialization, which needs identifying meaningful phrases for each class, random initialization simplifies the process by eliminating this requirement. This approach not only simplifies the process but also removes the dependency on the BERT model from our pipeline, thereby saving some computational resources.

# 4. Experiments

In this section, we present our implementation details, evaluation metrics, and datasets. We use a pretrained Grounding-DINO base model [33] for all experiments. For zero-shot detection, we employ text prompts that best describe the dataset classes (either single words or phrases). In our few-shot experiments on target datasets, we use multiple embeddings per class, with an ablation study showing that 4 embeddings per class yield good performance.

For all experiments, we initialize text embeddings randomly from a normal distribution and train these embeddings for 400 iterations with a batch size of 4 on an Nvidia A100 GPU. We use an initial learning rate of 2.0 with cosine decay and AdamW optimizer. Each image is resized by scaling its shorter side to 800 pixels while maintaining the aspect ratio, and we apply random horizontal flips and random cropping as training augmentations. For few-shot training, unless otherwise noted, we randomly select 1, 2, 4, 8, 16, and 24 images from the training set of each target dataset and evaluate on complete test set. For consistency in our experiments, 'shots' are referred to as the number of training images used. We perform 10 runs for each few-shot setup by sampling random few-shot training images and report mean and standard deviation scores.

We evaluate model performance using object detection metrics: Mean Average Precision (mAP) at Intersection over Union (IoU) thresholds of 50:95%, 50%, and 75%. Unless otherwise stated, mAP refers to mAP@50:95 in our paper. We benchmark the model on 8 datasets, as detailed in the following subsection.

## 4.1. Object Detection

**Crop-Weed Dataset [18]:** This dataset is designed to evaluate computer vision models for precision agriculture tasks. It contains top-down field images captured by an autonomous field robot in an organic carrot farm during various crop growth stages, with images taken when one or more true leaves are present. The dataset includes two classes: crop (carrot) and weed. This dataset presents unique challenges due to high similarity between weed plants and young carrot plants, as well as frequent partial occlusion of carrots by weeds, as shown in Figure 3.

First, we perform an ablation study to determine the optimal number of text embeddings per class. We use the same 4 and 8 training images and increase the number of text embeddings while keeping other parameters same. Table 1 shows that performance improves as we increase the number of text embeddings per class; however, it plateaus after 4 embeddings per class. We observe similar trends in both 4-shot and 8-shot settings. For consistency across experiments, we choose 4 embeddings per class for all datasets.

Next, we compare the performance of our few-shot approach against the zero-shot baseline. For the zero-shot setup, we tested various prompts for the carrot plant class, such as 'small carrot plant' and 'carrot plant.' However, the model frequently misclassifies instances, often detecting all instances simply as 'plant' or 'weed,' as shown in Figure 3. We report the zero-shot results for the prompt 'crop . weed' in Table 2a. From these results, we observe that the few-shot approach significantly outperforms the zero-shot method. With 24 images, our few-shot approach achieves an mAP of 43.0, outperforming zero-shot detection at 10.5.

Table 1. Comparison of mAP for different number of text embeddings per class on Crop-Weed dataset [18]. Here number of shot is number of training images used.

| Number of Embeddings | 4-Shot | | 8-Shot | |
|---|---|---|---|---|
| | mAP@(50:95) | mAP@50 | mAP@(50:95) | mAP@50 |
| 2 | 39.5 | 62.8 | 39.2 | 64.0 |
| 4 | 42.6 | 67.7 | 42.6 | 68.3 |
| 6 | 41.5 | 67.2 | 42.1 | 68.1 |
| 8 | 42.1 | 68.7 | 42.2 | 68.4 |
| 10 | 42.2 | 68.7 | 42.7 | 68.9 |

**BUP20 Dataset [55]:** This dataset includes images of sweet peppers across five classes: red, yellow, green, mixed red, and mixed yellow. It is challenging dataset due to significant occlusion by green leaves as shown in Figure 3.

For zero-shot, we used the prompt 'red pepper. yellow pepper. green pepper. mixed red pepper. mixed yellow pepper.' Table 3 compares zero-shot, and few-shot performance with varying training images. With one image, few-shot learning underperforms compared to zero-shot, likely because not all class instances are represented in a single image. However, increasing training images improves performance significantly. With 24 images, our few-shot approach achieves an mAP of 38.1, outperforming zero-shot detection at 21.7.

We also compare the performance of our few-shot approach with YOLOv11[20]. We use coco-pretrained YOLOv11-nano model and fine-tune the entire model parameters on the few training images for 100 epochs. Table 3 demonstrate that the few-shot approach significantly surpasses YOLOv11, achieving an mAP improvement of up to $\sim 24\%$ when trained on just 4 images.

**SB20 Dataset [2]:** This dataset contains RGB images of two classes: sugar beet and weeds. The images cover

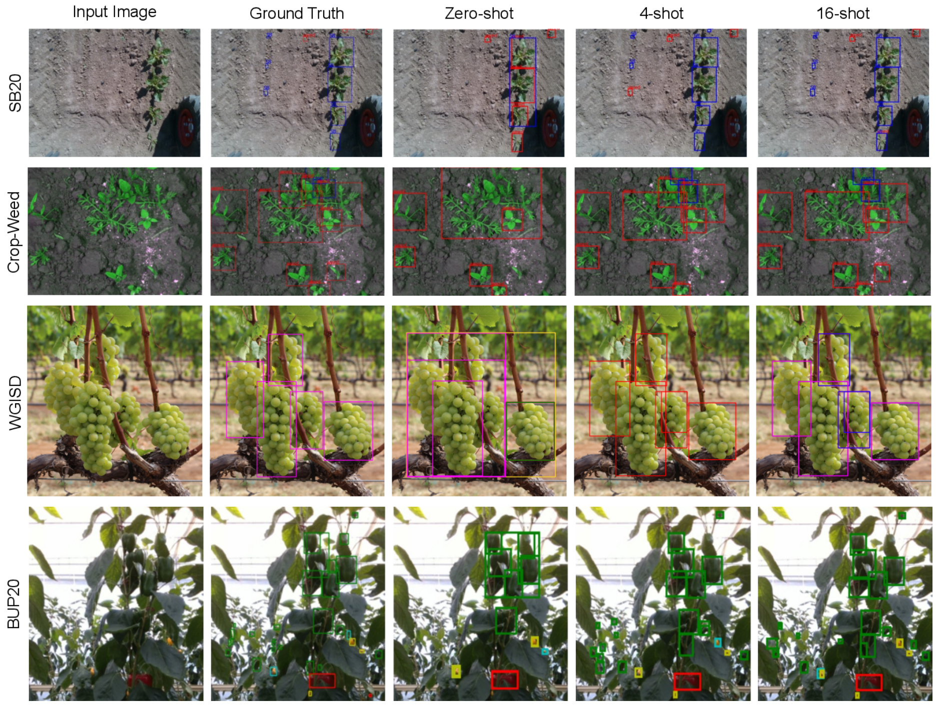


Figure 3. Figure compares zero-shot vs our few-shot approaches using Grounding-DINO on SB20[2], Crop-Weed[18], Grape Detection (WGISD)[47], and BUP20[55]. Zero-shot fails in cluttered/occluded environments, struggling to detect instances or distinguish similar classes. Our few-shot approach (4-shot and 16-shot) outperforms zero-shot on all datasets.

Table 2. Quantitative results on various datasets comparing our few-shot approach with zero-shot. Our few-shot method significantly outperforms zero-shot across all datasets, with performance improving as we increase the number of training images.

(a) Crop-Weed

| **Number of training images** | **mAP @(50:95)** | **mAP @50** | **mAP @75** |
|---|---|---|---|
| 0 | 10.5 | 26.8 | 6.6 |
| 1 | 22.7 ± 7.0 | 39.1 ± 12.4 | 24.0 ± 6.8 |
| 2 | 31.5 ± 4.1 | 53.0 ± 6.2 | 31.3±5.4 |
| 4 | 36.9 ± 3.3 | 61.9 ± 5.2 | 38.7±4.4 |
| 8 | 39.5 ± 1.6 | 64.6 ± 2.6 | 42.8±3.2 |
| 16 | 42.5 ± 2.1 | 68.0 ± 3.3 | 46.6 ± 2.8 |
| 24 | **43.0 ± 3.5** | **68.6 ± 4.9** | **48.7 ± 4.8** |

(b) SB20

| **Number of training images** | **mAP @(50:95)** | **mAP @50** | **mAP @75** |
|---|---|---|---|
| 0 | 26.5 | 38.6 | 28.9 |
| 1 | 19.6 ± 7.1 | 33.8 ± 12.2 | 19.6 ± 7.9 |
| 2 | 29.4 ± 7.7 | 50.3 ± 12.3 | 29.6 ± 8.4 |
| 4 | 37.7 ± 6.2 | 61.7 ± 8.2 | 38.3 ± 7.0 |
| 8 | 41.9 ± 5.8 | 67.8 ± 7.7 | 42.7 ± 6.5 |
| 16 | 45.6 ± 1.8 | 72.8 ± 2.3 | 46.9 ± 2.0 |
| 24 | **46.4 ± 2.0** | **73.3 ± 2.6** | **47.6 ± 2.2** |

(c) PhenoBench

| **Number of training images** | **mAP @(50:95)** | **mAP @50** | **mAP @75** |
|---|---|---|---|
| 0 | 12.6 | 23.2 | 12.4 |
| 1 | 27.8 ± 6.7 | 53.4 ± 9.3 | 25.4 ± 8.0 |
| 2 | 32.8 ± 1.4 | 61.4 ± 1.6 | 30.8 ± 2.2 |
| 4 | 35.3 ± 1.3 | 65.4 ± 1.5 | 33.8 ± 1.9 |
| 8 | 37.4 ± 0.5 | 67.9 ± 1.0 | 36.3 ± 0.7 |
| 16 | 37.3 ± 1.7 | 67.7 ± 2.2 | 36.0 ± 2.0 |
| 24 | **38.2 ± 1.0** | **68.7 ± 1.3** | **37.1 ± 1.4** |

(d) Insect Detection

| **Number of training images** | **mAP @(50:95)** | **mAP @50** | **mAP @75** |
|---|---|---|---|
| 0 | 16.0 | 25.0 | 19.1 |
| 1 | 6.5 ± 1.6 | 10.2 ± 2.3 | 6.8 ± 1.7 |
| 2 | 15.1 ± 2.1 | 23.2 ± 4.0 | 16.6 ± 2.4 |
| 4 | 20.8 ± 5.8 | 30.5 ± 8.2 | 23.5 ± 6.8 |
| 8 | 30.8 ± 4.8 | 45.0 ± 7.7 | 34.7 ± 5.1 |
| 16 | 39.0 ± 4.6 | 56.1 ± 6.3 | 44.0 ± 5.2 |
| 24 | **41.0 ± 5.0** | **59.0 ± 6.6** | **46.3 ± 5.7** |

(e) Grape Detection

| **Number of training images** | **mAP @(50:95)** | **mAP @50** | **mAP @75** |
|---|---|---|---|
| 0 | 3.8 | 6.1 | 4.3 |
| 1 | 2.9 ± 1.0 | 4.3 ± 1.2 | 3.0 ± 1.1 |
| 2 | 9.7 ± 2.9 | 14.5 ± 4.4 | 10.5 ± 3.3 |
| 4 | 22.6 ± 7.6 | 32.9 ± 11.2 | 24.7 ± 8.6 |
| 8 | 32.1 ± 8.0 | 46.5 ± 11.7 | 34.7 ± 8.7 |
| 16 | **35.8 ± 9.2** | **51.2 ± 13.5** | **39.0 ± 10.1** |
| 24 | 35.4 ± 7.0 | 49.9 ± 9.9 | 39.0 ± 7.9 |

(f) Wheat Head Detection

| **Number of training images** | **mAP @(50:95)** | **mAP @50** | **mAP @75** |
|---|---|---|---|
| 0 | 10.2 | 22.8 | 7.5 |
| 1 | 31.4 ± 4.1 | 74.0 ± 6.5 | 20.5 ± 4.6 |
| 2 | 34.8 ± 3.3 | 79.9 ± 4.2 | 23.5 ± 3.8 |
| 4 | 37.1 ± 2.6 | 83.3 ± 3.4 | 26.1 ± 3.0 |
| 8 | 37.9 ± 1.6 | 84.7 ± 1.8 | 26.6 ± 2.0 |
| 16 | 39.6 ± 0.8 | 86.4 ± 0.8 | 29.0 ± 1.1 |
| 24 | **40.1 ± 1.0** | **86.8 ± 0.9** | **29.9 ± 1.5** |

Table 3. Comparison of our few-shot approach against YOLOv11's performance on the BUP20 dataset [55], where the entire YOLOv11 model is trained using few training images.

| **Number** | **Few-shot** | | **YOLOv11** | |
|---|---|---|---|---|
| **of Images** | **mAP@(50:95)** | **mAP@50** | **mAP@(50:95)** | **mAP@50** |
| 0 | 21.7 | 32.2 | - | - |
| 1 | 17.5 ± 3.3 | 26.8 ± 4.4 | 2.9 ± 0.5 | 4.7 ± 0.7 |
| 2 | 23.1 ± 2.4 | 34.8 ± 3.5 | 3.5± 0.8 | 5.7± 1.2 |
| 4 | 28.1 ± 2.7 | 42.1 ± 3.5 | 4.6±0.5 | 7.6±0.9 |
| 8 | 32.1 ± 2.3 | 47.9 ± 3.5 | 12.5±0.8 | 19.8±1.2 |
| 16 | 36.5 ± 2.2 | 53.2 ± 3.0 | 18.4±1.7 | 29.1±2.5 |
| 24 | 38.1 ± 1.5 | 56.1 ± 2.1 | 21.8±2.0 | 34.3±2.7 |

a range of growth stages, natural world illumination conditions, and challenging occlusions. For zero-shot detection, we tested prompts like 'sugar beet' and 'plant,' but the model struggled to distinguish between sugar beet and weeds as shown in Figure 3.

In Table 2b, we use the prompt 'crop . weed' for the zero-shot experiments. The table shows that with one training image, few-shot learning underperforms compared to zero-shot. This is likely due to the diverse growth stages of sugar beet plants and randomly sampling one training image fails to capture the full variation across all growth stages. However, increasing training images improves performance significantly. With 24 training images, our few-shot approach achieves an mAP of 46.4, outperforming zero-shot detection at 26.5. Additionally, the standard deviation decreases as we increase the number of training images.

**Pheno Bench Dataset [62]:** This dataset comprises large high-resolution images captured with unmanned aerial vehicles (UAV) of sugar beet fields under natural lighting conditions over multiple days. For our experiment, we focus on detecting individual leaves. This task is challenging due to the cluttered nature of the leaves and varying growth stages.

For zero-shot detection, we use the 'leaf instance' prompt. However, as shown in Figure 1, the model tends to detect entire rows or plants rather than individual leaves. Table 2c compares zero-shot and few-shot performance. We find that even with one training image, the few-shot approach outperforms zero-shot. With 24 images, our few-shot method achieves an mAP of 38.2, significantly surpassing zero-shot detection at 12.6.

**Insect Detection Dataset [54]:** This dataset includes high-resolution images of various insects across six classes: fly, honeybee, hover fly, shadow, wasp, and other insect. It is challenging due to the small size of insects and similar-looking species, making differentiation difficult.

For zero-shot text prompting, this dataset presents challenges as the 'other insect' class encompasses numerous species, complicating the creation of descriptive prompts that capture all variations within the class while distinguishing it from others. For quantification, we use the class names directly for prompting. Table 2d compares zero-shot and few-shot performance. We observe that few-shot detection with 1 or 2 images performs less effectively than zero-shot, as most dataset images contain insects from one or two classes, limiting model exposure to other species. However, increasing training images significantly improves performance; with 24 images, our few-shot approach achieves an mAP of 41.0, outperforming zero-shot detection at 16.0.

**Grape Detection Dataset [47]:** It provides instances of five different grape varieties (Chardonnay, Cabernet Franc, Cabernet Sauvignon, Sauvignon Blanc, Syrah) captured under field conditions. These instances exhibit variance in grape pose, illumination as well as genetic and phenological characteristics such as shape, color, and compactness.

For zero-shot prompting, we prepend 'grape' to every class name. Figure 3 provides a qualitative comparison between zero-shot and few-shot detection. We observe that the zero-shot model struggles to detect individual clusters, often grouping them into one instead. Table 2e shows that for 1 and 2 training images, few-shot performance is suboptimal as these images typically contain only one or two classes each. However, at higher training numbers, our few-shot approach significantly outperforms zero-shot; for instance, with 16 training images, it achieves 35.8 mAP compared to 3.8 for zero-shot.

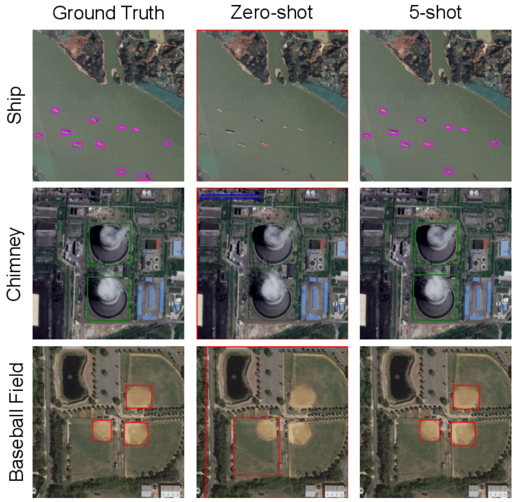


Figure 4. Qualitative comparison between zero-shot and few-shot approaches on DIOR [27] dataset, demonstrating that our few-shot method achieves significantly better results.

**Wheat Head Dataset [12]:** This dataset consists of images of wheat fields and contains a single class: 'wheat spike head'. It is designed for counting wheat spike heads. The task presents significant challenges due to the presence of clutter and occluded heads.

For zero-shot detection, we used the prompt 'wheat spike head'. As shown in Figure 1, the zero-shot model detects only a few heads and fails to detect occluded ones. Table 2f shows that the few-shot approach significantly outperforms zero-shot detection, even with just one training image. With 24 training images, the few-shot method achieves 40.1 mAP compared to 10.2 for zero-shot.

**Remote Sensing Dataset [27]:** Object detection in optical remote sensing (DIOR) dataset is a widely used benchmark for evaluating few-shot models in remote sensing. Collected from Google Earth, it spans over 80 countries, offering extensive variations in environmental conditions such as weather, seasons, and imaging quality. Our experiments are conducted across all splits of the dataset, each containing five classes. Following the experimental setup outlined in [29], we use the same images for each n-shot experiment. In this set up, the number of shots refers to the number of instances per class.

Figure 4 presents the qualitative results on DIOR Split-1, which contains 5 classes: baseball field, basketball court, bridge, chimney, and ship. For zero-shot detection, we use class names as prompts; Grounding-DINO achieves an mAP of 22.8. We compare our few-shot approach with state-of-the-art (SOTA) methods in Table 4. Our approach significantly outperforms previous methods: for 3-shot detection, we achieve an mAP of 40.0, compared to the previous SOTA method's 31.69. Consistent improvements are observed across other splits: on Split-2 for 3-shot, our approach achieves an mAP of 38.5, compared to the previous SOTA's 14.5; and on Split-3 for 3-shot, we attain an mAP of 30.0, outperforming the previous SOTA's 18.85.

Table 4. Results of our few-shot approach and state-of-the-art few-shot model performance on DIOR dataset [27] (split-1).

| Method | 3-shot | 5-shot | 10-shot | 20-shot |
|---|---|---|---|---|
| FSCE [56] | 27.91 | 28.60 | 33.05 | 37.55 |
| SAE-FSDet [34] | 28.80 | 32.40 | 37.09 | 42.46 |
| G-FSDet [65] | 27.60 | 29.89 | 34.86 | 37.49 |
| GE-FSOD [29] | 31.69 | 34.88 | 38.02 | 43.08 |
| **Ours** | **40.0** | **42.3** | **48.3** | **47.6** |

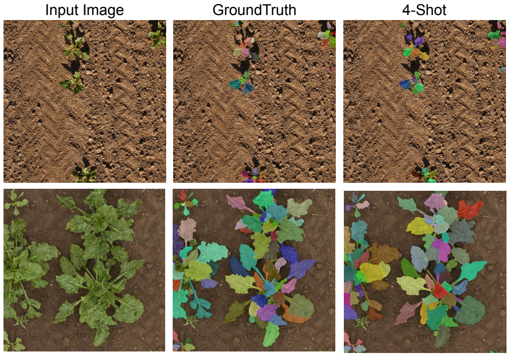


Figure 5. Instance segmentation on PhenoBench dataset [62] using our few-shot Grounding-DINO and SAM2.

### 4.2. Instance Segmentation

We extend our few-shot object detection approach to instance segmentation. Following [44], we use the SAM2 model [39] as our foundation, using the few-shot predicted bounding boxes from our Grounding-DINO based detection method as prompts for SAM2. For experimentation, we utilize the PhenoBench dataset [62], where we perform leaf instance segmentation. Figure 5 demonstrates qualitative results on this dataset. The few-shot approach performs well in instance segmentation, particularly for plants at early growth stages, where leaves are less cluttered and the model achieves good performance. However, challenges arise with larger plants, as overlapping leaves lead to some segmentation errors due to occlusion. In terms of mAP with mask IoU, our model achieves 31.8 with just one training image and improves to 42.3 with eight training images.

### 4.3. Discussion

Our few-shot approach significantly outperforms the zero-shot method using Grounding-DINO model across diverse agricultural datasets, even with very limited training images and steps. Through our experiments (Table 2), we observe that the zero-shot Grounding-DINO model excels at detecting fruits and plants when they are visually sparse but struggles in cluttered or occluded scenarios, often failing to detect individual instances or grouping multiple instances together. Additionally, for multi-class fine-grained distinctions, such as crop-weed differentiation, the zero-shot approach proves inadequate in distinguishing between similar looking classes. These challenges stem from difficulties in formulating effective text prompts for individual instance detection and intra-class differentiation. Furthermore, datasets like insects, where multiple classes are grouped into one category, pose additional hurdles in crafting appropriate text prompts.

Our few-shot approach addresses the challenges of prompting and demonstrates significant improvements over the zero-shot method with minimal training images. Our experiments reveal that using just one training image for multi-class datasets, generally underperforms compared to zero-shot, as a single image often does not contain all class instances. However, as we increase the number of training images, the performance of our few-shot approach improves significantly. When comparing our few-shot approach to YOLO (Table 3), where the entire model parameters are fine-tuned, we observe that our method outperforms YOLO when the number of training images is limited. However, with all training images, YOLO's fully fine-tuned model performs better. Our future work would explore adding more trainable parameters to Grounding-DINO, such as through low-rank adapters, for scenarios with a larger number of training data. Additionally, when benchmarked against SOTA few-shot methods in remote sensing using standard experimental setups, our approach demonstrates significant improvements. This likely stems from the strong pre-training of Grounding-DINO on extensive datasets. Furthermore, the pre-trained Grounding-DINO demonstrates robust adaptability across diverse datasets-from lab environments to satellite imagery-using minimal few-shot training examples. This versatility underscores its capacity to generalize seamlessly across modalities without domain-specific retraining, positioning it as a powerful tool for automating annotation pipelines in precision agriculture. While Grounding-DINO incurs notable computational costs, our proposed method can be seamlessly integrated into YOLO-based frameworks like YOLO-World[10] and YOLOE[59] for efficient few-shot detection tasks.

## 5. Conclusion

In this paper, we investigate the application of the Grounding-DINO model for agricultural object detection tasks. We identified significant challenges in manual text prompting for agriculture-specific scenarios and developed an efficient few-shot learning method that eliminates reliance on text prompts. Our experimental results demonstrate superior performance across diverse datasets, particularly in challenging environments with occluded objects and fine-grained distinctions. Looking ahead, our work opens new avenues for research in methods for efficiently adapting open-set object detection models pretrained on large datasets and developing few-shot learning approaches tailored to agricultural datasets. The insights gained from this study contribute to the broader goal of advancing AI applications in agriculture, ultimately supporting more sustainable and efficient farming practices.

## References

[1] Iftikhar Ahmad, Yayun Yang, Yi Yue, Chen Ye, Muhammad Hassan, Xi Cheng, Yunzhi Wu, and Youhua Zhang. Deep learning based detector yolov5 for identifying insect pests. *Applied Sciences*, 12(19):10167, 2022. 1

[2] Alireza Ahmadi, Michael Halstead, and Chris McCool. Towards autonomous crop-agnostic visual navigation in arable fields. *CoRR*, 2021. 2, 5, 6

[3] Simone Antonelli, Danilo Avola, Luigi Cinque, Donato Crisostomi, Gian Luca Foresti, Fabio Galasso, Marco Raoul Marini, Alessio Mecca, and Daniele Pannone. Few-shot object detection: A survey. *ACM Comput. Surv.*, 54(11s), 2022. 2

[4] Mar Ariza-Sentís, Sergio Vélez, Raquel Martínez-Peña, Hilmy Baja, and João Valente. Object detection and tracking in precision farming: A systematic review. *Computers and Electronics in Agriculture*, 219:108757, 2024. 1

[5] Mar Ariza-Sentís, Sergio Vélez, Raquel Martínez-Peña, Hilmy Baja, and João Valente. Object detection and tracking in precision farming: a systematic review. *Computers and Electronics in Agriculture*, 219:108757, 2024. 2

[6] Chetan M Badgujar, Alwin Poulose, and Hao Gan. Agricultural object detection with you only look once (yolo) algorithm: A bibliometric and systematic literature review. *Computers and Electronics in Agriculture*, 223:109090, 2024. 1, 2

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1, 3, 4

[8] Jian-Wen Chen, Wan-Ju Lin, Hui-Jun Cheng, Che-Lun Hung, Chun-Yuan Lin, and Shu-Pei Chen. A smartphone-based application for scale pest detection using multiple-object detection methods. *Electronics*, 10(4):372, 2021. 1

[9] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in neural information processing systems*, 34:17864–17875, 2021. 1

[10] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16901–16911, 2024. 1, 8

[11] Bini Darwin, Pamela Dharmaraj, Shajin Prince, Daniela Elena Popescu, and Duraisamy Jude Hemanth. Recognition of bloom/yield in crop images using deep learning models for smart agriculture: A review. *Agronomy*, 11(4):646, 2021. 1

[12] Etienne David, Franklin Ogidi, Daniel Smith, Scott Chapman, Benoit de Solan, Wei Guo, Frederic Baret, and Ian Stavness. Global wheat head detection challenges: Winning models and application for head counting. *Plant Phenomics*, 5:0059, 2023. 1, 2, 7

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019. 3

[14] Akshay Raj Dhamija, Manuel Günther, Jonathan Ventura, and Terrance E. Boult. The overlooked elephant of object detection: Open set. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1010–1019, 2020. 2

[15] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14084–14093, 2022. 1

[16] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 1

[17] Guangxing Han and Ser-Nam Lim. Few-shot object detection with foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28608–28618, 2024. 2

[18] Sebastian Haug and Jörn Ostermann. A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks. In *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part IV 13*, pages 105–116. Springer, 2015. 2, 5, 6

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[20] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLO, 2023. 5

[21] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1780–1790, 2021. 1

[22] Baoshuo Kan, Teng Wang, Wenpeng Lu, Xiantong Zhen, Weili Guan, and Feng Zheng. Knowledge-aware prompt tuning for generalizable vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15670–15680, 2023. 2

[23] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014. 1

[24] Anand Koirala, Kerry B Walsh, Zhenglin Wang, and Cheryl McCarthy. Deep learning–method overview and review of use for fruit detection and yield estimation. *Computers and electronics in agriculture*, 162:219–234, 2019. 1

[25] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Ui-

jlings, Stefan Popov, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2017. 1

[26] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 2

[27] Ke Li, Gang Wan, Gong Cheng, Liqiu Meng, and Junwei Han. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS journal of photogrammetry and remote sensing*, 159:296–307, 2020. 2, 7, 8

[28] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. 1, 3

[29] Hui Lin, Nan Li, Pengjuan Yao, Kexin Dong, Yuhan Guo, Danfeng Hong, Ying Zhang, and Congcong Wen. Generalization-enhanced few-shot object detection in remote sensing. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. 7, 8

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 1

[31] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 3, 4

[32] Lihao Liu, Juexiao Feng, Hui Chen, Ao Wang, Lin Song, Jungong Han, and Guiguang Ding. Yolo-uniow: Efficient universal open-world object detection. *arXiv preprint arXiv:2412.20645*, 2024. 2

[33] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024. 1, 2, 5

[34] Yanxing Liu, Zongxu Pan, Jianwei Yang, Bingchen Zhang, Guangyao Zhou, Yuxin Hu, and Qixiang Ye. Few-shot object detection in remote sensing images via label-consistent classifier and gradual regression. *IEEE Transactions on Geoscience and Remote Sensing*, 2024. 8

[35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 3

[36] Hongpeng Pan, Shifeng Yi, Shouwei Yang, Lei Qi, Bing Hu, Yi Xu, and Yang Yang. The solution for cvpr2024 foundational few-shot object detection challenge. *arXiv preprint arXiv:2406.12225*, 2024. 2

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2

[38] Nitiyaa Ragu and Jason Teo. Object detection and classification using few-shot learning in smart agriculture: A scoping mini review. *Frontiers in Sustainable Food Systems*, 6, 2023. 2

[39] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1, 8

[40] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 2

[41] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[42] Shaoqing Ren. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 1

[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2015. 2

[44] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024. 1, 8

[45] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019. 3, 4

[46] Thiago T. Santos, Leonardo L. de Souza, Andreza A. dos Santos, and Sandra Avila. Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. *Computers and Electronics in Agriculture*, 170:105247, 2020. 2

[47] Thiago T Santos, Leonardo L De Souza, Andreza A dos Santos, and Sandra Avila. Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. *Computers and Electronics in Agriculture*, 170: 105247, 2020. 2, 6, 7

[48] Walter J. Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, 2013. 2

[49] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural

machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015. 3

[50] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 1

[51] Cheng Shi and Sibei Yang. Edadet: Open-vocabulary object detection using early dense alignment. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15724–15734, 2023. 1

[52] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020. 2

[53] Hyeon-Seok Sim and Hyun-Chong Cho. Enhanced deepsort and strongsort for multicattle tracking with optimized detection and re-identification. *IEEE Access*, 13:19353–19364, 2025. 2

[54] Maximilian Sittinger, Johannes Uhler, Maximilian Pink, and Annette Herz. Insect detect: An open-source diy camera trap for automated insect monitoring. *Plos one*, 19(4):e0295474, 2024. 2, 7

[55] Claus Smitt, Michael Halstead, Tobias Zaenker, Maren Bennewitz, and Chris McCool. Pathobot: A robot for glasshouse crop phenotyping and intervention. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2324–2330. IEEE, 2021. 2, 5, 6

[56] Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. Fsce: Few-shot object detection via contrastive proposal encoding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7352–7362, 2021. 8

[57] Zhe Tang, Jiajia Lu, Zhengyun Chen, Fang Qi, and Lingyan Zhang. Improved pest-yolo: Real-time pest detection based on efficient channel attention mechanism and transformer encoder. *Ecological Informatics*, 78:102340, 2023. 2

[58] Adar Vit, Guy Shani, and Aharon Bar-Hillel. Length phenotyping with interest point detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1

[59] Ao Wang, Lihao Liu, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yoloe: Real-time seeing anything. *arXiv preprint arXiv:2503.07465*, 2025. 8

[60] Liqiong Wang, Jinyu Yang, Yanfu Zhang, Fangyi Wang, and Feng Zheng. Depth-aware concealed crop detection in dense agricultural scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17201–17211, 2024. 2

[61] Ya-Hong Wang and Wen-Hao Su. Convolutional neural networks in computer vision for grain crop phenotyping: A review. *Agronomy*, 12(11):2659, 2022. 1

[62] Jan Weyler, Federico Magistri, Elias Marks, Yue Linn Chong, Matteo Sodano, Gianmarco Roggiolani, Nived Chebrolu, Cyrill Stachniss, and Jens Behley. Phenobench: A large dataset and benchmarks for semantic image interpretation in the agricultural domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1, 2, 7, 8

[63] Size Wu, Wenwei Zhang, Sheng Jin, Wentao Liu, and Chen Change Loy. Aligning bag of regions for open-vocabulary object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15254–15264, 2023. 1

[64] Zhimeng Xin, Shiming Chen, Tianxu Wu, Yuanjie Shao, Weiping Ding, and Xinge You. Few-shot object detection: Research advances and challenges. *Information Fusion*, 107: 102307, 2024. 2

[65] Tianyang Zhang, Xiangrong Zhang, Peng Zhu, Xiuping Jia, Xu Tang, and Licheng Jiao. Generalized few-shot object detection in remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 195:353–364, 2023. 8

[66] Yang Zhang, Chenglong Song, and Dongwen Zhang. Deep learning-based object detection improvement for tomato disease. *IEEE Access*, 8:56607–56614, 2020. 2

[67] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825, 2022. 2

[68] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2