Abstract

Effective object detection in precision agriculture requires understanding minimum dataset requirements, yet this remains undetermined for arable crops seedling detection. This study investigates the minimum dataset size and quality needed to achieve benchmark performance ($R^2 = 0.85$) across different object detection paradigms. We systematically evaluated many-shot models (YOLOv5, YOLOv8, YOLO11, RT-DETR), few-shot (CD-ViTO), and zero-shot (OWLv2) approaches using orthomosaic imagery of maize seedlings, while also implementing a handcrafted algorithm as baseline. Models were tested with varying dataset sizes, quality levels, and training sources (in-domain vs out-of-distribution). Results demonstrate that no out-of-distribution trained model achieved benchmark performance, while in-domain trained models reached the benchmark with 60-130 annotated images, depending on architecture. Transformer-mixed models (RT-DETR) required fewer samples (60) than CNN-based models (110-130), but showed different sensitivities to annotation quality reduction. Models maintained benchmark performance with 65-90% of original annotation quality. Neither few-shot nor zero-shot approaches met benchmark requirements despite their recent advances. These findings provide practical guidance for efficiently developing maize seedling detection systems, emphasizing that successful deployment requires in-domain training data, with minimum requirements dependent on model architecture.

2.1.1 Introduction

2.1.1.1 The Problem of Plant Counting

Plant counting is a critical operation in precision agriculture, plant breeding, and agronomical evaluation. Accurate plant counts can provide valuable information for both farmers and researchers. This task was often performed manually by human operators. Today, this process can be automated by the use of computer vision algorithms. To validate a method for counting, it is critical to set a benchmark for accuracy. A benchmark can be defined by the accuracy of manual counting, international standards, or by comparison with other already accepted methods. Accuracy of plant manual counting depends on human performance, so variable, but often taken as golden sample. According to the European Plant Protection Organization, the benchmark for acceptance is a coefficient of determination (R^2) of 0.85 when compared to manual counting [8]. This corresponds to a Root Mean Square Error ($RMSE$) of approximately 0.39. Also scientific literature mention a R^2 value of 0.85 with ground truth (manual counting) as a benchmark for acceptance [85].

Literature shows that benchmark for acceptance can be achieved with computer vision object detection [57, 22, 17] or regression models [56, 2]. A superiority of object detection over regression models in terms of accuracy was found [85]. Object detection is also more versatile than regression models, because, object detection model inference on georeferenced orthomosaics delivers plants ge-

ographical coordinates, not only density per area. The validation of this ability rely on metrics as Intersection over Union (IoU), Average Precision (AP), and Average Recall [54] rather than the coefficient of determination. Identification supports are usually bounding boxes, which are rectangles that enclose the object of interest, but they can also be points or other kind of geometries. Counting of plants is then commonly performed by counting the number of geometries that enclose the objects of interest (the plant) in a image area. Georeferenced orthomosaics are images created through aerial photogrammetry, a process that involves capturing overlapping georeferenced images taken by nadiral view picturing georeferenced ground control points, and performing bundle adjustment to form a single, seamless image [46]. Georeferenced orthomosaics are scaled and oriented to a geographical coordinate system. It implies that the pixel coordinates corresponds to geographical coordinates and are projectables in a metric system. Using georeferenced orthomosaics in seedling counting makes object detection easier, because the metric scale can be used to locate the objects in the image and prospective variance is reduced to zero.

2.1.1.2 Case study: Maize Seedling Counting

Plant counting is affected, as many object detection application fields, from data scarcity: public datasets are rare and often not suitable for the task because of the large environmental variability and their images are not orthorectified or scale is unknown. Even so, some useful dataset for training a plant counting object detector can be

found in public repositories [32]. These dataset may come as a part of a scientific study or with poor technical specifications. Selecting a specific crop at a specific growth stage can reduce the variability that the model should learn and make it possible to better study the other variables that can affect the dataset size and quality requirements. For this study we choose grain maize seedlings (Zea mays L.) at the V3-V5 (BBCH 13-15) growth stage [61], because it is the most represented plant in scientific [22, 57] and not scientific [5, 4] open datasets from aerial photogrammetry. Maize is a commodity crop that is widely grown in the world and is the most important crop in the world by production [25]. Grain maize seedlings in that stage are easy to count because of their low overlapping and fixed intra-row and inter-row spacing, differently by silo-maize seedling that is seeded with extremely low inter-row spacing. This particular seedling configuration that makes grain maize a good candidate for object detection, is shared with other crops that are seeded in rows with inter-row spacing such as sunflower (Helianthus annuus L.) or sugarbeet (Beta vulgaris L.).

2.1.1.3 Object Detection approaches

The development of object detection algorithms has evolved from not machine learning methods, here named handcrafted methods (HC), to modern deep learning-based (DL) techniques. Today, all the state-of-the-art object detection methods are based on DL models. Nevertheless, HC methods are still used in some cases [22, 29]. Most of modern DL object detection uses convolutional neural net-

works (CNN) [48] based frameworks (e.g., Faster R-CNN [3], YOLO [7]) or Transformer [72] based approaches (e.g., DETR [19]) or mixtures of both approaches. The main difference between CNN-based and Transformer-based models is the way they process the image. CNN-based models process the image in a grid-like fashion (convolutions), while Transformer-based models process the image as a sequence of patches (attention mechanisms) [23]. On common benchmarks as COCO [54], PASCAL VOC [38], and ImageNet [1], Transformer-based models have shown to be more accurate than CNN-based models [84] . CNN-based models are still widely used in object detection, because they are more efficient in processing small images and have a lower computational cost [43]. However, when fine-tuning with scarce data, Transformer-based object detectors generally perform better than CNN-based detectors, provided they are pretrained on large datasets [66, 51, 12].

To represent the categories and compare performance between pure-CNN and Transformer-mixed architectures that are effectively used as object detectors in real plant counting applications, YOLOv5 and YOLOv8 can be taken as pure-CNN architecture representatives for their large use in agriculture [15]. Their large diffusion in agriculture applications is justified by the fact that they leverage good precision and the low need in terms of dataset size in respect other CNN architectures [71, 53, 81]. Real-Time-DETR (RT-DETR) is a recent Transformer-mixed architecture that outperforms YOLOv5 and YOLOv8 [83]. YOLO11 has been recently proposed as a Transformer-mixed YOLO architecture that outperforms RT-DETR on COCO dataset [44].

Recently, zero-shot and few-shot object detection have emerged as promising paradigms to alleviate the need for large annotated datasets. While traditional object detection models (many-shots object detectors) require extensive labeled data for training, few-shot and zero-shot object detection aim to detect novel objects with little to no labeled examples respectively. The term ”shot” refers to the number of annotations used to train the model over all the images. Each shot corresponds to an individual, that in the case of few-shots, is used to prototype the object of interest. Zero and few-shots approaches often leverage feature transfer or meta-learning components to generalize across classes under extreme data scarcity [52]. This approach reduces the annotation burden and is especially beneficial in domains where collecting exhaustive training data is impractical. Zero-shot object detection detects new categories without any training samples by leveraging semantic relationships or contextual information learned from known classes [16]. Few-shot approaches optimize models to learn quickly from a handful of labeled examples [35]. Meta learning is a common approach in few-shot object detection [6, 78, 27, 80], while Cross-Domain Few-Shot Object Detection (CD-FSOD) has recently surpassed this approach by leveraging domain adaptation techniques [68]. DE-ViT [79] and CD-ViTO [28] are the latest models that have shown promising results in few-shot object detection. Zero-shot actual state-of-the-art object detector models are YOLO-World [40] , OWLv2 [62] , and Grounding DINO [55]. OWLv2 (Open-World Localization v2) is an zero-shot object detector that represented a significant evolution in open-vocabulary detection capabilities.

2.1.1.4 Dataset Size and Quality Requirements

As already mentioned, the performance comparison between object detection models is usually based on differences in metrics such as AP, calculated on standard datasets that are not representative of the agricultural application field. Many studies have been conducted on object detection for plants in open field [17, 30, 37, 42, 45, 49, 50, 57, 56, 58, 59, 73, 74, 82], but few have argued or focused on dataset minimum requirements for training a robust plant object detector [22, 13]. Some study focused on few-shots approach for plant counting [74, 11], but only two specifically accounting on few-shots method for maize seedling counting [41, 75]. Unfortunately, the lonely two studies evaluating few-shot performance on maize seedling counting by orthomosaics do not achieve the benchmark and do not clearly specify the number of shots used. Also no zero-shot benchmark for maize seedling counting has been set yet, and no research on this application has been done yet.

This critical research gap presents significant challenges for practitioners in precision agriculture who must decide how much data to collect and annotate for effective maize seedling detection. Without systematic evaluation of minimum dataset requirements across different detection approaches (many-shot, few-shot, and zero-shot), it remains unclear whether resource-intensive manual annotation can be reduced or eliminated. Furthermore, the agricultural domain's unique characteristics: variable environment, and plant phenotype, may fundamentally alter the data requirements compared to general computer vision benchmarks. Determining these requirements

would provide practical guidance for implementing object detection in agricultural workflows while optimizing the trade-off between annotation effort and detection performance.

Even if no research has been made in order to minimize or set a benchmark for the dataset size and quality required to train a maize seedling object detector, it is well known that the performance of a DL model is directly related to the amount of data used for training [70] and its quality [10]. Dataset size and quality predictability in DL has been proven to be addressable with empirical approaches [34, 60]. As already mentioned, model architecture is critical in dataset requirements as different models may require different dataset sizes and qualities to achieve the same performance [64, 18]. Backbone is pivotal on downstream tasks as object detection [24]. The importance of using a domain specific backbone has been proven also for plant/leaves segmentation on orthomosaics [67], but backbone training is still prohibitive for many applications. No backbone weights specialized on agricultural orthomosaics has been published yet, as for the most of the specialized domains. Even if a so specialized backbone exists, some concern will came out about its out-of-distribution generalization capability [33]. So, because of limited resources, today only the use of a general backbone is possible in practical sense [36], even if a decreasing in dataset size requirements is expected with a domain specific backbone because of the out-of-distribution generalization [31]. Another factor that can affect the dataset size and quality requirements other than the already cited model architecture and use of pre-trained backbones, is the training data augmentation strategy. Some studies proposed high-

tly computationally expensive data augmentation such as trainable data augmentation [21] or data augmentation with generative adversarial networks [14]. Other studies proposed to use less computationally expensive data augmentation strategies [69, 20] and someone even proved that random image augmentation can provide equivalent results to more expensive technics [63]. As also image augmentation can lead to prohibitive computational costs, the use of less computationally expensive data augmentation strategies is recommended for fine-tuning to downstream tasks [69].

Nevertheless, the most important factor affecting DL training is the dataset source [70]. It has been observed that using training samples from the same dataset as the inference (in-domain dataset) dramatically increases accuracy and reduce the need for a large dataset in respect to collecting training samples from other datasets (out-of-distribution dataset) [22, 13]. From comparison of the studies here mentioned, dataset source seems to be the most important factor in determining the dataset size and quality needed to achieve the benchmark for acceptance.

2.1.1.5 Study Aim

This study aims to determine the minimum dataset size and quality required to train a object detection model for identifying maize seedlings in georeferenced orthomosaics achieving the benchmarks set by international organizations and recognized in scientific literature. Here, the dataset size and quality are respectively defined as the amount of annotated images in the training set, and the accu-

racy of the annotations. Also the effect of training dataset source will be evaluated in this study. Models architecture and size will be taken into account, as long as the object detection downstream task is concerned. After setting the dataset size and quality minimum requirements for many-shots object detectors, we will evaluate if new zero-shot and few-shot object detection models can achieve the same benchmark for acceptance with less data and annotations. We will also discuss the need for HC method in a DL object detection pipeline.

2.1.2 Materials and Methods

2.1.2.1 Datasets

The datasets used in this study to train the object detection models for maize seedlings counting are nadiral or supposedly nadiral images of maize seedlings at the V3-V5 growth stage or estimated so. The V3-V5 growth stage is defined by the BBCH scale as the stage where the third to fifth leaf is unfolded and the plant is 15-30 cm tall [61].

This study uses two dataset sources as training sets: the out-of-distribution dataset (OOD) and the in-domain dataset (ID). The ID datasets are from the same source as the testing dataset, while the OOD datasets are not. The OOD datasets are composed of images from scientific literature [56, 22] and from internet repositories [5, 4]. The ID datasets were collected during this study. This ID dataset

creation consisted in capturing nadiral images of three study areas with a Phantom 4 Pro v2.0 (DJI, Shenzhen, China) drone equipped with its default series RGB camera @ 10 m AGL (above the ground level). The number of images captured depends on the study area size that was about 2 hectares for ID_1 location and about 1 ha for the other two. For each location, an orthomosaic was created using a photogrammetric software. Bundle adjustment error was estimated as 38 mm using the ground control points surveyed by GNSS operating in VRS-NRTK mode. The orthomosaics were generated with an average ground sampling distance of 5 mm/pixel in the WGS84/UTM 32 N reference system.

The OOD scientific datasets consist of tiles of georeferenced orthomosaics of maize seedlings from scientific literature. The OOD internet datasets consist of RGB images of maize seedlings from internet repositories. The ID datasets were collected during this study and consist of tiles of georeferenced orthomosaics of maize seedlings of known scale. The OOD scientific datasets and the ID datasets are composed of tiles of georeferenced orthomosaics of known scale, while the OOD internet datasets are simple RGB images of unknown scale. All the OOD datasets came with annotations, while the ID datasets were manually annotated. The OOD dataset annotation are rectangular bounding boxes centering on an individual plant stem. ID dataset annotation was done during this study by an agronomist by observing the entire orthomosaic on a Geographical Information System (GIS) environment, with the tiles grid overlapping the orthomosaic to focus on target tiles without losing the surrounding context, so without losing bordering plants. An-

notations were created as squared bounding boxes of size length equal to the minimum distance between two plants in the row, with each box centered on an individual seedling stem.

To make the two kind of dataset comparables we chose to rescale the images to a scale of 0.005 m/pixel where the scale was known (scientific OOD and ID datasets), obtaining orthomosaics of different sizes. All the orthomosaics were then cropped to 224*224 pixels tiles. This tile size was selected because at 5 mm/pixel resolution it covers 1.12×1.12 meters of field area. Given that typical grain maize inter-row distance is 0.75 meters, this size enables capturing approximately two rows per tile, which is optimal for row pattern identification in the HC algorithm and provides sufficient context for object detection models. This particular image size was also chosen as a standard from AlexNet [47] as it should be compatible with most of the object detection architectures. The annotations were rescaled and cropped where needed. Figure 1 shows a sample for each dataset. Each ID dataset has 20 tiles to be used as testing dataset, while other 150 tiles are used as training dataset.

2.1.2.2 Handcrafted object detector

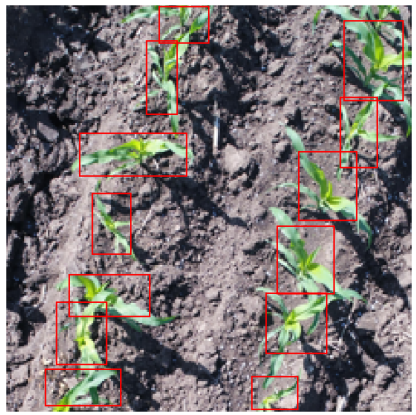
Like other works [22, 29, 56] we wrote an HC algorithm to get annotated tiles from the orthomosaics, basing it on agronomical knowledge and color thresholding. Hue, saturation and value (HSV) color space was used here to threshold the image, to get green pixels, but other color spaces can be used. For the execution of this algorithm, the following graphical and agronomical parameters must be

Table 1: Summary of Datasets Used in the Study

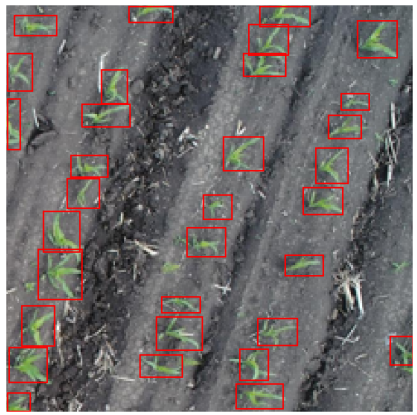
Dataset	Phenological Stage	Train Size	Test Size
OOD Scientific			
DavidEtAl.2021 [22]	V3	182 tiles	N/A
LiuEtAl.2022 [56]	V3	596 tiles	N/A
OOD Internet			
OOD_int_1 [5]	V3	216 tiles	N/A
OOD_int_2 [4]	V5	174 tiles	N/A
ID			
ID_1	V3	150 tiles	20 tiles
ID_2	V3	150 tiles	20 tiles
ID_3	V5	150 tiles	20 tiles

(a) DavidEtAl.2021

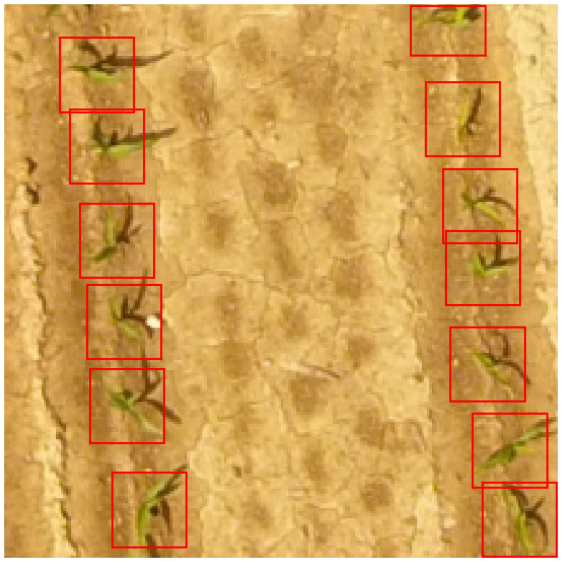
(b) LiuEtAl.2022



(c) Internet Maize stage V3

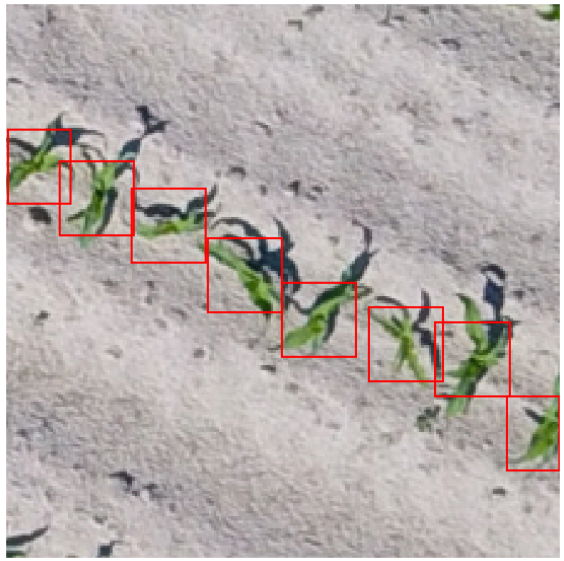


(d) Internet Maize stage V5



(e) ID_1

(f) ID_2



(g) ID_3

Figure 1: Sample images from each dataset used in the study. The top row shows out-of-distribution datasets: **(a)** DavidEtAl.2021, **(b)** LiuEtAl.2022, **(c)** Internet Maize stage V3, and **(d)** Internet Maize stage V5. The bottom row shows in-domain datasets: **(e)** ID_1, **(f)** ID_2, and **(g)** ID_3.

set: color minimum and maximum thresholds (color threshold), the minimum and maximum leaf area for plant (leaf area range), the minimum distance between plants on rows (intra-row distance), and the distance between rows (inter-rows distance). The algorithm is expected to works: on orthomosaics of maize seedlings at the V3-V5 growth stage, with low weeds infestation, with rows having roughly the same angle with meridian and distance between them.

The algorithm is divided in two sequential parts that form a detection-verification pipeline. The first part, named HC1 algorithm 1, performs initial plant detection by thresholding pixels within the specified color range, identifying connected regions, and filtering them based on expected leaf area. HC1 outputs region polygons representing potential plants, but typically includes many false positives due to its simple color-based approach. To address this limitation, we implemented a second process named HC2 (algorithm 2 and algorithm 3) that applies agronomical knowledge of field structure. HC2 filters the HC1 output by verifying that detected plants form proper row patterns with expected intra-row and inter-row spacing. It uses RANSAC to identify linear alignments of plants and validates that these alignments match expected field geometry (consistent row slope and spacing). Only tiles where HC2 confirms the expected number and arrangement of plants are retained for the final dataset. This two-stage approach enables automated extraction of high-confidence annotations from the orthomosaics.

Algorithm 1 H1

Require: $tiles$ ▷ Orthomosaic tiles
Require: col_range ▷ Color space thresholds
Require: $leaf\_area\_range$ ▷ Leaf area range in pixels
Ensure: $plants$ ▷ List of polygons

function connected_components($binary\_image$) [77]
    return $regions$
for $tile$ in $tiles$ **do**
    $mask \leftarrow \{p \in tile \mid color(p) \in col\_range\}$
    $regions \leftarrow connected\_components(mask)$
    $plants \leftarrow \{region \mid region \in regions \wedge region.area \in leaf\_area\_range\}$
return $plants$

2.1.2.3 Deep Learning object detectors

We chose them for the considerations made in Introduction section 2.1.1.3. We used here the Ultralytics implementation of these models because the implementation is open-source [39], and it makes it possible to tune parameters size coherently across all tested architectures.

All model training and inference was performed on a workstation equipped with an Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz, 64.0 GB RAM, and an NVIDIA RTX A5000 GPU with 24GB VRAM. The computational constraints influenced certain experimental design choices, such as batch size and precision settings.

For all the many-shots models we used the same hyperparameters and augmentations as the library default, with the following exceptions:

- batch size: 16 (increased from default 8 to maximize GPU uti-

Algorithm 2 H2: Part I - Row Detection

Require: $observations$ ▷ (List of centroids, RanSaC models)
Require: $intra - row\_dist$ ▷ Minimum distance between plants
Require: $inter - row\_dist$ ▷ Minimum distance between rows
Require: $mean\_slope$ ▷ Mean slope of the rows in respect meridian
Ensure: $objects$ ▷ List of centroids or polygons

function region_centroids($regions$) ▷ Get the centroids of the regions
    return $centroids$
function agglomerate_regions($regions, min\_dist$) ▷ Agglomerate regions
    $centroids \leftarrow \{region.centroid \mid region \in regions\}$
    $clusters \leftarrow$ HierarchicalClustering$(centroids, threshold = min_dist, metric =$ euclidean$)$
    $clust_cen \leftarrow \{$mean$(centroids_i) \mid$ for each cluster $i \in clusters\}$
    return $clust\_cen$
function extract_ransac_line($points, min\_dist$) [26]
    return $best\_inliers, best\_model$
function process_tiles($intra - row\_dist$)
    $observations \leftarrow \{\}$
    $plants \leftarrow$ HC1$(tiles)$
    for $tile$ in $tiles$ **do**
        $regions \leftarrow plants[tile]$
        $centroids \leftarrow$ region_centroids$(regions)$
        $clust\_cen \leftarrow$ agglomerate_regions$(regions, intra - row_dist)$
        $inlier\_points, model \leftarrow$ extract_ransac_line$(clust_cen, intra - row_dist)$
        $line\_length \leftarrow$ get_line_length$(model)$
        $expected\_number\_of\_plants \leftarrow \frac{line\_length}{intra\_row\_dist}$
        if $inlier\_points \equiv expected\_number\_of\_plants$ **then**
            $observations[tile] \leftarrow (clust\_cen, inlier\_points, model)$
    return $observations$

Algorithm 3 H2: Part II - Row Verification

```
function Filter_observations_by_slope(observations)
    filtered_observations ← {}
    for tile ∈ observations do
        slope ← observations[tile]['model']
        if model.slope ≈ mean_slope then
            filtered_observations[tile] ← observations[tile]
    return filtered_observations
function process_observations(observations, inter − row_dist, intra − row_dist)
    objects ← {}
    for tile ∈ observations do
        tile_centers ← observations[tile]['clust_cen']
        first_row_centers ← observations[tile]['inlier_points']
        first_row_model ← observations[tile]['model']
        centers ← {p | p ∈ tile_centers ∧ p ∉ first_row_centers}
        second_row_centers, second_row_model ← extract_ransac_line(centers, intra − row_dist)
        line_length ← get_line_length(second_row_model)
        expected_number_of_plants ← line_length / intra_row_dist
        if second_row_model.slope ≈ first_row_model.slope then
            if abs(second_row_model.intercept − first_row_model.intercept) ≈ inter − row_dist then
                if second_row_centers ≡ expected_number_of_plants then
                    objects[tile] ← (first_row_centers, second_row_centers)
    return objects
function main
    observations ← process_tiles(intra − row_dist)
    MEAN_SLOPE ← mean(observations['model'])
    observations ← Filter_observations_by_slope(observations, MEAN_SLOPE)
    objects ← process_observations(observations, inter − row_dist)
    return objects
```

lization while maintaining stable gradients)

- maximum training epochs: 200 (extended from default 100 to ensure convergence with small datasets)
- maximum training epochs without improvement: 15 (increased from default 10 for early stopping to allow longer plateau exploration)
- precision: mixed (to balance training speed and numerical accuracy)

The default augmentations from the Ultralytics library include random scaling (±10%), random translation (±10%), random horizontal flip (probability 0.5), HSV color space augmentation (hue ±0.015, saturation ±0.7, value ±0.4), and mosaic augmentation. These augmentations were selected to reflect potential variations in field conditions without introducing unrealistic distortions.

The training dataset was composed of the OOD or ID training dataset tiles. For the dataset size testing, all the annotations were used, while for the dataset quality testing a percentage of the annotations per image was selected and used.

To test the few-shot approach we trained CD-ViTO with multiple model sizes. The size of this model is determined by the backbone used, which can be ViT-S, ViT-B, or ViT-L [65]. We used the implementation of CD-ViTO provided by the authors [28]. In the context of this study a 'shot' correspond to an image with a single annotated plant. We used 1, 5, 10, 30, and 50 shots to train the model. The

shots were randomly selected from the ID manually labeled dataset, then a random annotation was selected from the same image to be used as prototype. All the combinations of shots and ViT backbone were tested on the ID test dataset tiles.

For testing the zero-shot approach we used OWLv2. We took this architecture as zero-shot object detector example as it is the most stable state-of-the-art model for this task [62, 55]. For test OWLv2 we used the implementation of the transformer library [76] with the parameters published by the authors. We tested the encoder sizes ViT-B/16, ViT-L/14 with the following three pre-training strategies:

- **Base models**: Trained using self-supervised learning with the OWL-ST method, which generates pseudo-box annotations from web-scale image-text datasets
- **Fine-tuned models**: Further trained on human-annotated object detection datasets
- **Ensemble models**: Combining multiple weight-trained versions to balance open-vocabulary generalization and task-specific performance

For all the OWLv2 variants, we tested multiple text prompts to describe maize seedlings, ranging from simple terms (”maize”, ”seedling”) to more descriptive phrases (”aerial view of maize seedlings”, ”corn seedlings in rows”). The complete list of eleven prompts is the following:

- ”maize”

- ”seedling”
- ”plant”
- ”aerial view of maize seedlings”
- ”corn seedlings in rows”
- ”young maize plants from above”
- ”crop rows with corn seedlings”
- ”maize seedlings with regular spacing”
- ”top-down view of corn plants”
- ”agricultural field with maize seedlings”
- ”orthomosaic of corn plants in rows”

All the combinations (here named model settings) of encoder size, pre-training strategy, and text prompt were tested on the ID test dataset tiles.

table 2 shows the architectures used in the study with the parameter size specifics.

Table 2: Summary of Tested Architectures and Model Sizes[1]

Architecture	Shots	n[2]	s[2] or S	m[2] or B	l[2] or L	x[2]
YOLOv5	many	1.9	7.2	21.2	46.5	86.7
YOLOv8	many	3.2	11.2	25.9	43.7	68.2
YOLO11	many	4.0	12.5	28.0	50.0	75.0
RT-DETR	many	-	-	-	60.0	80.0
CD-VITO	few	-	22.0[3]	86.0[4]	307.0[5]	-
OWLv2	zero	-	-	86.0[4]	307.0[5]	-

[1] Values represent millions of parameters

[2] Model size variants stand for nano (n), small (s), medium (m), large (l), and extra-large (x)

[3] ViT-S (Small) backbone

[4] ViT-B (Base) backbone

[5] ViT-L (Large) backbone

2.1.2.4 Minimum dataset size and quality modelling

In order to investigate the minimum size and quality of the dataset required to train a robust object detection model for maize seedlings counting, we conducted a series of experiments where the above mentioned DL models were recursivelly fitted with increasing dataset size and quality. For many-shots models we consider a training dataset split of 10% validation and 90% training, while for few-shots the number of shots determined the amount of training samples. Zero-shots relied only on descriptions in natural language of the objects to be detected. For what concerns only the dataset size eval-

uation, for many-shots models we considered sizes from 10 to 150 images in 15 steps of 10 images, while for few-shots models we considered 1, 5, 10, 30, and 50 shots. For what concerns the dataset quality, we evaluated the annotation quality by reducing the number of annotations per image from 100% to 10% in 10 steps of 10% while keeping the dataset size constant.

For all the models we evaluated the relationship between dataset size or quality and model performance using R^2 and mAP, respectively for plant countig and plant detection. Wheter R^2 provided values below -1 we also considered $RMSE$ as metric for counting. MAPE was considered for few-shots and zero-shots models only to evaluate the quality of the annotations produced by the prediction of these models. We list here the metrics formulas for clarity:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{1}$$

where y_i is the ground truth count for the i-th image, $\hat{y}_i$ is the predicted count, and $\bar{y}$ is the mean of all ground truth counts. R^2 ranges from $-\infty$ to 1, with 1 indicating perfect prediction, 0 indicating that the model predictions are no better than simply predicting the mean, and negative values indicating that the model performs worse than predicting the mean.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

where $RMSE$ measures the average magnitude of prediction errors in the original units (number of plants).

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{3}$$

where MAPE measures the percentage error relative to the actual values, providing a scale-independent measure of accuracy. It is expressed as a percentage, with lower values indicating lower percentage of false positive or false negative. Thus it was reported as an index of the quality of the annotations. Note that MAPE is only calculated for cases where $y_i \neq 0$ to avoid division by zero. It is particular useful for counting as testing tiles never have zero plants.

For object detection performance, we used the standard COCO evaluation metric:

$$mAP = \frac{1}{|IoU|}\sum_{t \in IoU} \text{AP}_t \tag{4}$$

where mAP (mean Average Precision) is calculated at a single IoU (Intersection over Union) threshold of 0.5. AP at the IoU threshold

is the area under the precision-recall curve for detections that meet that IoU threshold criterion.

To test the predictability minimum dataset size and quality required to train a robust (achieving benchmark) object detector for maize seedlings counting through empirical models, we test the logarithmic, arctan and algebraic root functions to fit the dataset size or quality versus performance relationships as suggested by previous studies [60]. For clarity we list here the functions tested:

$$\text{Logarithmic: } f(x) = a\ln(x) + b \tag{5}$$

$$\text{Arctan: } f(x) = a\arctan(bx) + c \tag{6}$$

$$\text{Algebraic Root: } f(x) = ax^{1/b} + c \tag{7}$$

For the model fits to dataset size versus performance relationships, we evaluated multiple fitting functions and selected the one with the highest goodness-of-fit:

$$GoF = R^2_{\text{fit}} = 1 - \frac{\sum_{i=1}^{n}(y_i - f(x_i))^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{8}$$

where y_i is the observed metric (either R^2 or mAP), $f(x_i)$ is the fitted value at dataset size x_i, and $\bar{y}$ is the mean of the observed metrics.

All the trained models were tested on the testing dataset tiles with the SAHI method [9]. The SAHI method slices the testing image into smaller overlapping segments (patches) of the same size as the training tiles and then tests the model on each of them. The model outputs from each patch are then merged by non-maximum suppression and cropped by the original tile extension. The use of such a method is justified by the fact that the model is trained on tiles and the testing dataset is composed by tiles, but the real application is on orthomosaics, so the same object can be present in more than one tile in a cutted (and occluded) way. The SAHI method overcomes this problem ensuring all the possible objects are evaluated by the model as a whole. Thus it is expected to give a better performance in respect to the use of the single tile as input for the model. The prediction were then thresholded by a list of confidence score thresholds to get the plant count. All the metrics were computed for different score thresholds for all the models to evaluate the model performance at different confidence levels. The values to thresholds bounding boxes score were: 0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.29, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99. The highest R^2 value whithin the thresholds was considered as the model performance for that experiment.

2.1.3 Results

2.1.3.1 Handcrafted object detector

To evaluate the HC object detector as a training dataset extractor from the ID dataset, we measure the amount of annotated tiles that the HC algorithm can extract from the orthomosaics and the accuracy it delivers in comparison to handmade annotations. Table 3 shows the performance of the HC object detector on the ID datasets by enumerating metrics and successfully annotated tiles. The metrics were computed on the testing dataset tiles.

Table 3: HC Object Detector Performance

Dataset	R^2	$RMSE$	$MAPE$	mAP	**tiles**	**dataset %**
ID_1	0.95	0.12	9%	0.87	1184	7.8%
ID_2	0.93	0.11	12%	0.81	279	4.2%
ID_3	0.87	0.18	16%	0.73	158	1.8%

Overall the HC object detector performed well on the ID datasets, with R^2 values above 0.85 for all the datasets. The $RMSE$ values were below 0.2, while the mAP values were above 0.7. The MAPE values were below 20% for all the datasets. The HC algorithm was able to extract a significant amount of annotated tiles from the orthomosaics, with a percentage of the dataset ranging from 1.8% to 7.8%. In nominal scale the number of tiles successfully annotated

by the HC algorithm was not constant, but always over 150 tiles, so we took this minimum amount as maximum dataset size for the many-shots training.

2.1.3.2 Many-shots object detectors

OOD training

The OOD scientific datasets ”DavidEtAl.2021” and ”LiuEtAl.2022” were tested singularly and in combination in the experiment named ”scientific OOD”. The OOD internet datasets ”internet OOD” were tested singularly and in combination with the OOD scientific datasets in the experiment named ”All OOD”. Each model and OOD dataset combination was tested on the testing dataset tiles of the three ID datasets.

None of the dataset combinations reached the benchmark R^2 value of 0.85 with any model. The coefficients of determinations and the root mean square errors for all the OOD experiments are shown in Figure 2. The Goodness-of-fit (GoF) values for the R^2 values were always low (below 0.2) for all the metrics. The lowest $MAPE$ value was slightly less than 20%. For these same models the mAP values were the highest, with the best model being YOLOv8n with the LiuEtAl.2022 dataset. No particular model size seems to provide better results with respect to the others, neither the increasing dataset size seems to drive a model size performance trend. As no model achieved the benchmark, no study was done on the dataset quality requirements to achieve such benchmark.

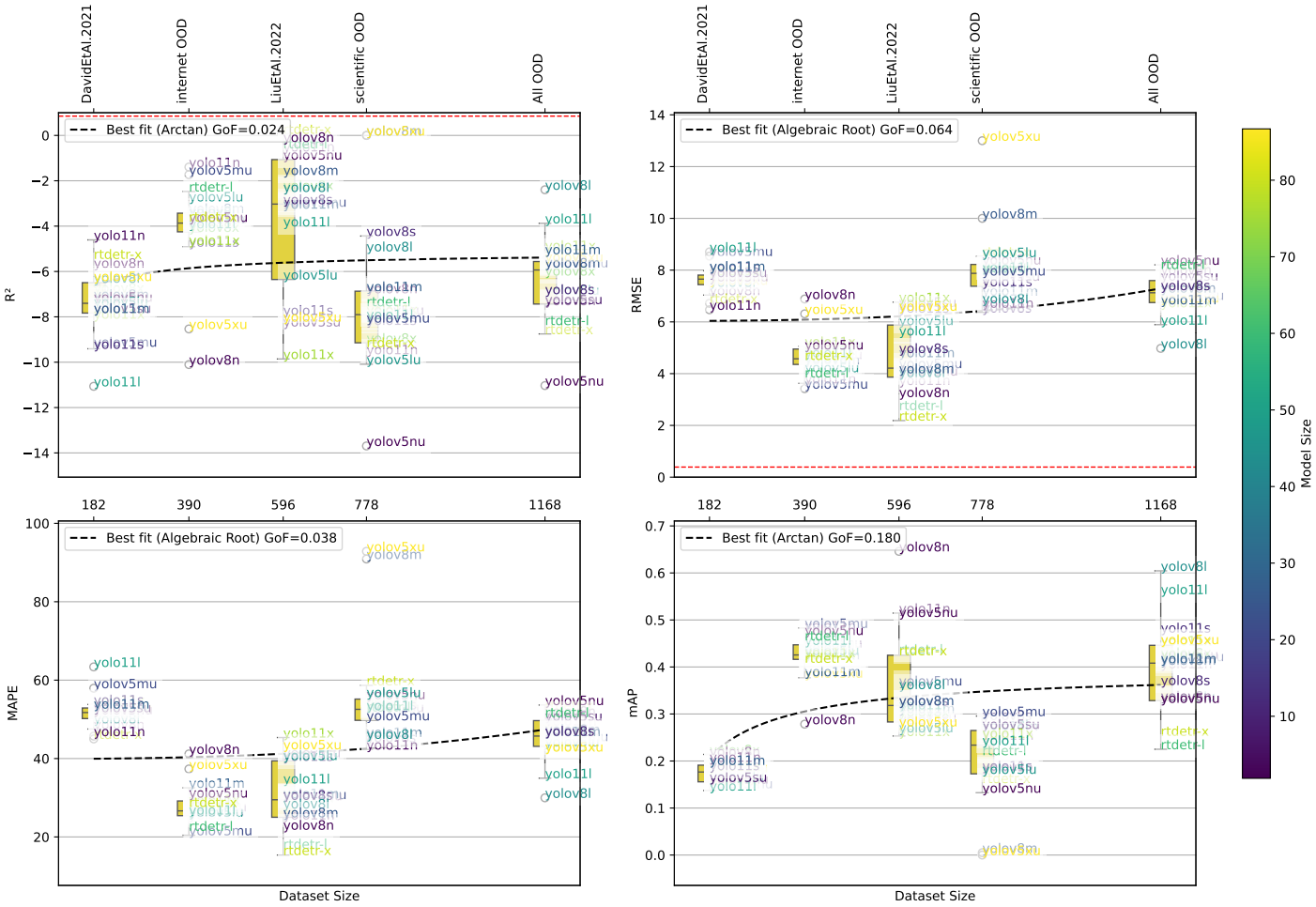


Figure 2: Performance of the many-shots object detection models trained on the different out-of-distribution (OOD) datasets. Subplots represent: R², $RMSE$, MAPE, and mAP respectively at the right top, left top, left bottom, and right bottom. Each subplot contains the boxplots positioned at the corresponding dataset size values and indicating the distribution of all the models prediction metric values for each dataset. Each data point is annotated with the , colored according to the model size. Benchmark thresholds are indicated with red dashed horizontal lines for R² (0.85) and $RMSE$ (0.39). Best fit lines for each metric are plotted using different fitting functions (logarithmic, arctan, and algebraic root), indicated with black dashed lines. GoF values and best model are shown in the legend. A secondary x-axis at the top of each subplot shows the dataset names corresponding to the dataset sizes.

ID training

The relationship between ID training dataset size and model performance was evaluated for all model architectures and sizes as shown in Figures 3, 4, 5 and 6. The dataset quality was tested later, taking the combination of model architecture, model size and training dataset size that achieved the benchmark and retraining that model while reducing the amount of annotations for each tile. The R^2 values of the counting and the mAP values for all models were regressed against the dataset size using a logarithmic, root or arctan model. The best fitting whitin them was selected for each model and metric and the GoF was calculated. A high GoF value indicates that model performance is highly predictable by dataset size. Conversely, a poor GoF could indicate that other variables play a more important role in determining model performance, or that the chosen dataset size interval is too narrow to achieve a good fit.

For the combinations of model-architecture/dataset-size that achieved the benchmark, the minimum dataset quality required to achieve the benchmark was evaluated as shown in Figure 7. The minimum dataset quality was determined by identifying the quality percentage where both the empirical model prediction and the entire confidence interval of the performance metrics remained above the benchmark threshold.

Within YOLO models, YOLOv5n, YOLOv5s and YOLOv8n achieve the benchmark R^2 value of 0.85 with 130, 130 and 110 samples, respectively, considering the dataset sizes where all three model performances were above 0.85 R^2 and the logarithmic model predicted

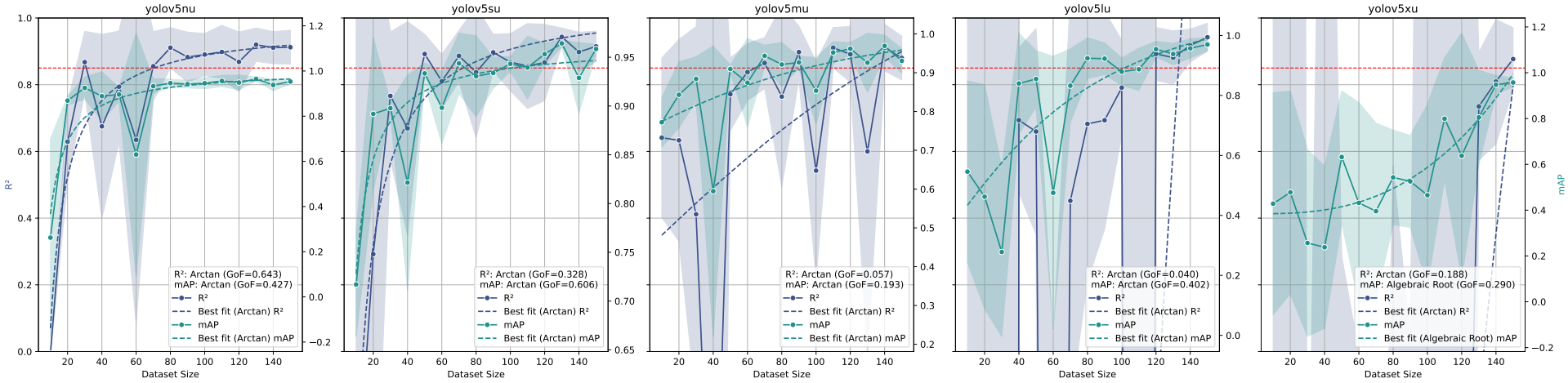


Figure 3: Relationship between dataset size and model performance for YOLOv5 trained and tested on ID datasets. Each subplot represents a different parameters size of the model, increasing from the left to the right. The x-axis represents the dataset size, while the left and right y-axis represents the R^2 and mAP values respectively. The solid lines represent the mean values, while the dashed lines indicate the logarithmic fit. The shaded area around the solid lines represents the confidence interval (standard deviation) of R^2 or mAP. The red dashed horizontal line represents the benchmark R^2 value of 0.85. The legend shows the goodness of fit (GoF) for both R^2 and mAP.

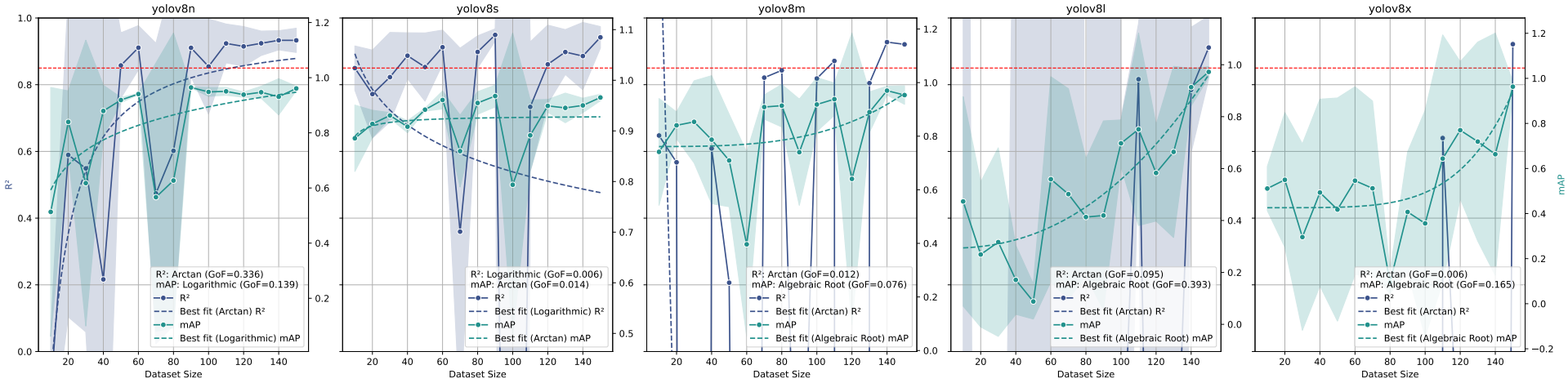


Figure 4: Relationship between dataset size and model performance for YOLOv8 trained and tested on ID datasets. Each subplot represents a different parameters size of the model, increasing from the left to the right. The x-axis represents the dataset size, while the left and right y-axis represents the R^2 and mAP values respectively. The solid lines represent the mean values, while the dashed lines indicate the logarithmic fit. The shaded area around the solid lines represents the confidence interval (standard deviation) of R^2 or mAP. The red dashed horizontal line represents the benchmark R^2 value of 0.85. The legend shows the goodness of fit (GoF) for both R^2 and mAP.

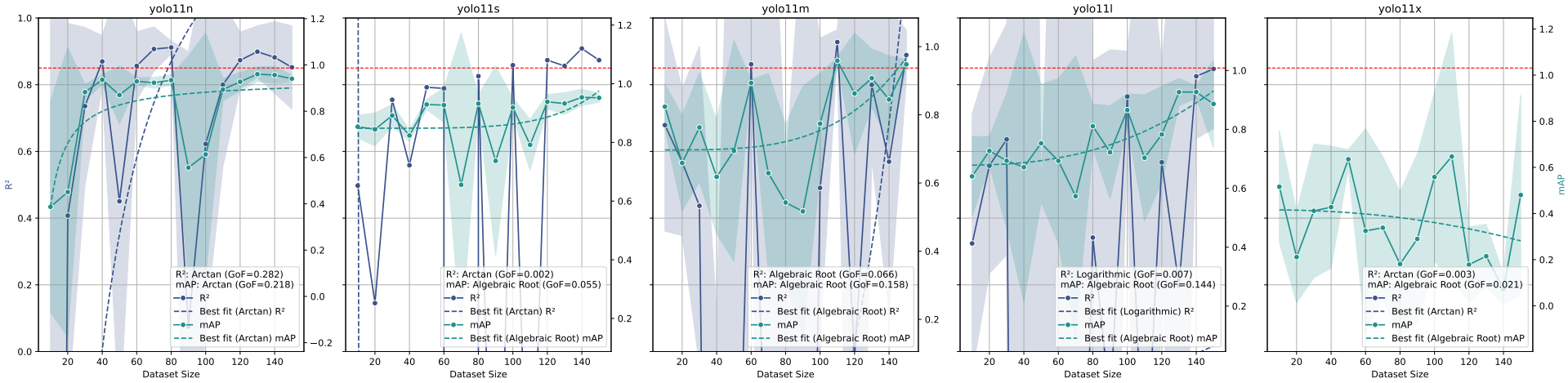


Figure 5: Relationship between dataset size and model performance for YOLO11 trained and tested on ID datasets. Each subplot represents a different parameters size of the model, increasing from the left to the right. The x-axis represents the dataset size, while the left and right y-axis represents the R^2 and mAP values respectively. The solid lines represent the mean values, while the dashed lines indicate the logarithmic fit. The shaded area around the solid lines represents the confidence interval (standard deviation) of R^2 or mAP. The red dashed horizontal line represents the benchmark R^2 value of 0.85. The legend shows the goodness of fit (GoF) for both R^2 and mAP.

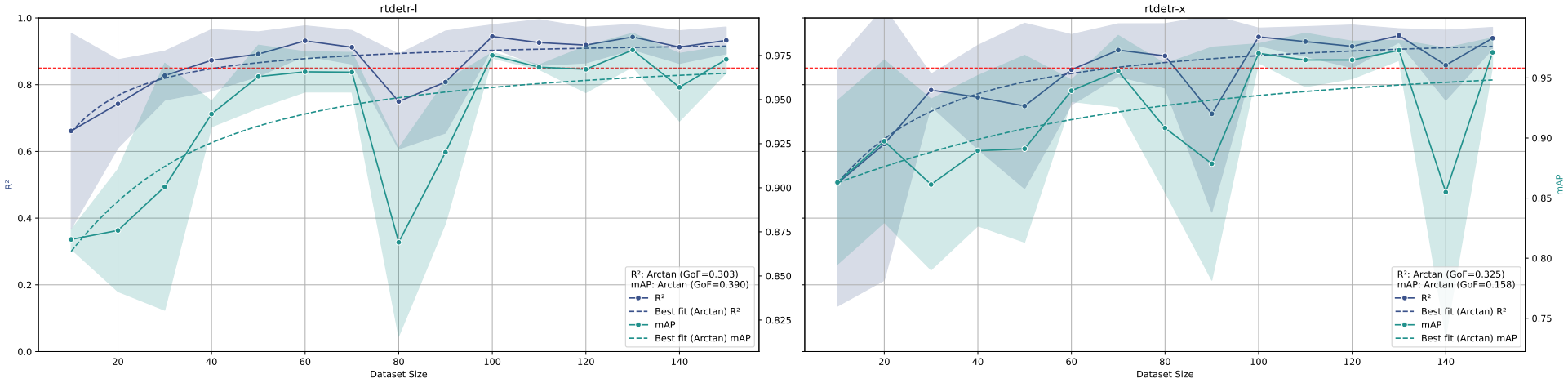


Figure 6: Relationship between dataset size and model performance for RT-DETR trained and tested on ID datasets. Each subplot represents a different parameters size of the model, increasing from the left to the right. The x-axis represents the dataset size, while the left and right y-axis represents the R^2 and mAP values respectively. The solid lines represent the mean values, while the dashed lines indicate the logarithmic fit. The shaded area around the solid lines represents the confidence interval (standard deviation) of R^2 or mAP. The red dashed horizontal line represents the benchmark R^2 value of 0.85. The legend shows the goodness of fit (GoF) for both R^2 and mAP.

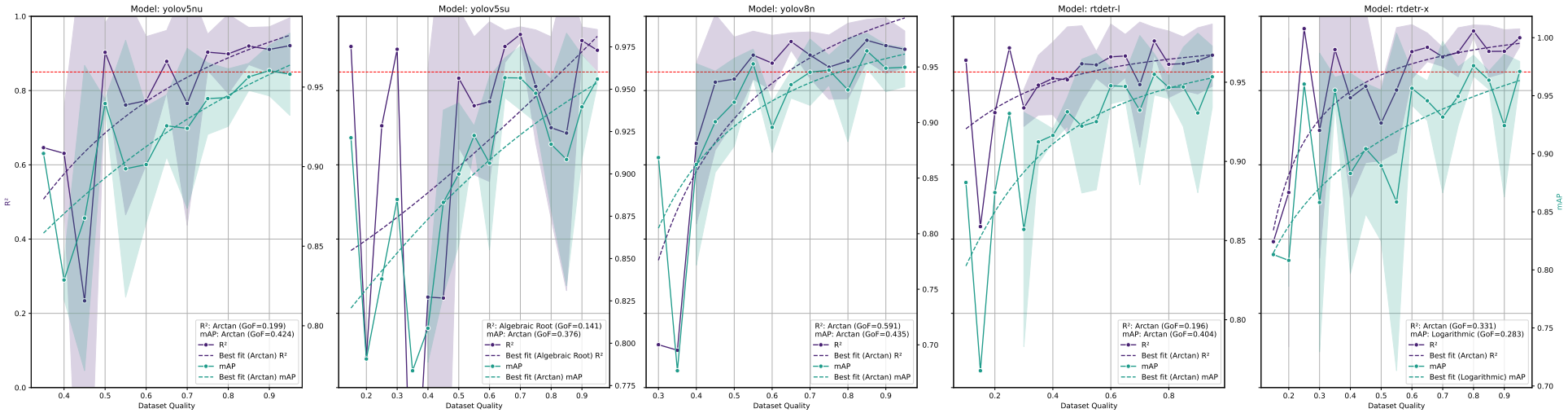


Figure 7: Relationship between dataset quality and model performance for all object detection models that achieved the benchmark. The x-axis represents the dataset quality, while the left y-axis represents the R^2 values. The red dashed horizontal line represents the benchmark R^2 value of 0.85. The legend in the lower right corner of the subplot shows the goodness of fit (GoF) for R^2.

over-benchmark values for that dataset size. RT-DETR L and RT-DETR X achieve the benchmark R^2 value of 0.85 with 60 and 100 samples respectively, with the same assumptions as for YOLO models. For these models the GoF was above 0.3, while for the models that did not reach the benchmark R^2 value the GoF was always below this value. The mAP seems to follow the same trend as the R^2 values. All the models show a clear trend of increasing R^2 and mAP values as the dataset size increases, as expected. It is also clear that increasing number of parameters and model complexity for mostly CNN-like models (YOLOs) leads to increasing need for dataset size. For the mostly transformer-like models (RT-DETRs) it is not that clear, also because of the low amount of model parameter sizes tested. The confidence interval reduction as a function of the dataset size indicates that variability in performance decreases significantly as dataset size increases for all models. Taking into account the dataset quality in the same way as done for the dataset size, both quality tests and quality models achieved the benchmark with 85%, 90%, 85% and 65% of the original dataset quality for YOLOv5n, YOLOv5s, YOLOv8n and RT-DETR X, respectively. RT-DETR L did not achieve the benchmark for any dataset quality reduction tested.

2.1.3.3 Few-shots object detectors

The few-shots models were evaluated against the established benchmarks (R^2 of 0.85 and $RMSE$ of 0.39) using the metrics R^2 and $RMSE$ because none of the models reached these benchmarks.

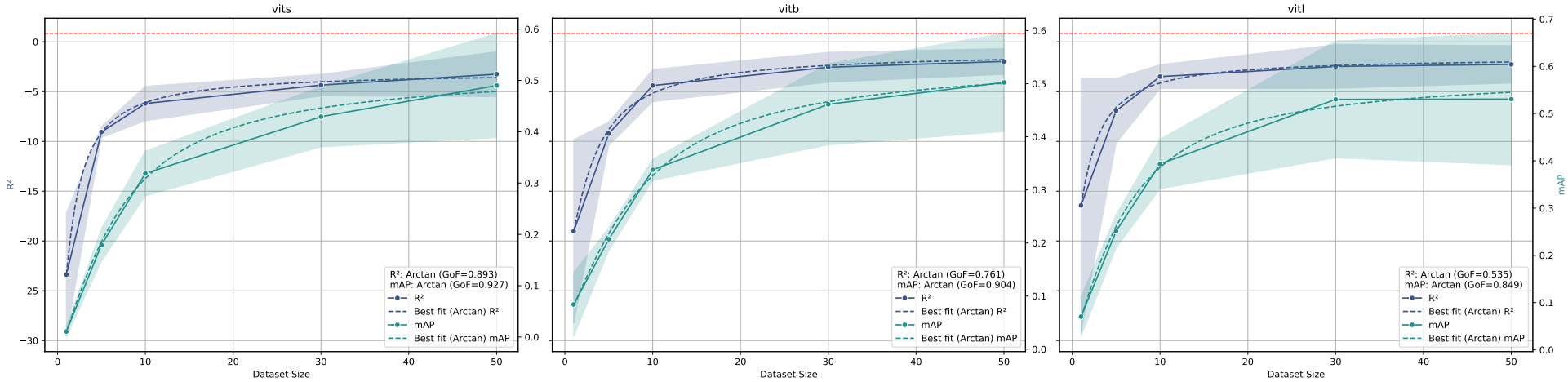


Figure 8: The figure shows the relationship between shots and model performance for the CD-ViTO model trained and tested on ID datasets. The x-axis represents the number of shots. The solid lines represent the mean values, while the dashed lines indicate the shots amount/metric prediction model. The left and right y-axis represents the R^2 and mAP values respectively. The red dashed horizontal line represents the benchmark R^2 value of 0.85. The combined legend in the lower right corner of each subplot shows the goodness of fit (GoF) for both R^2 and mAP.

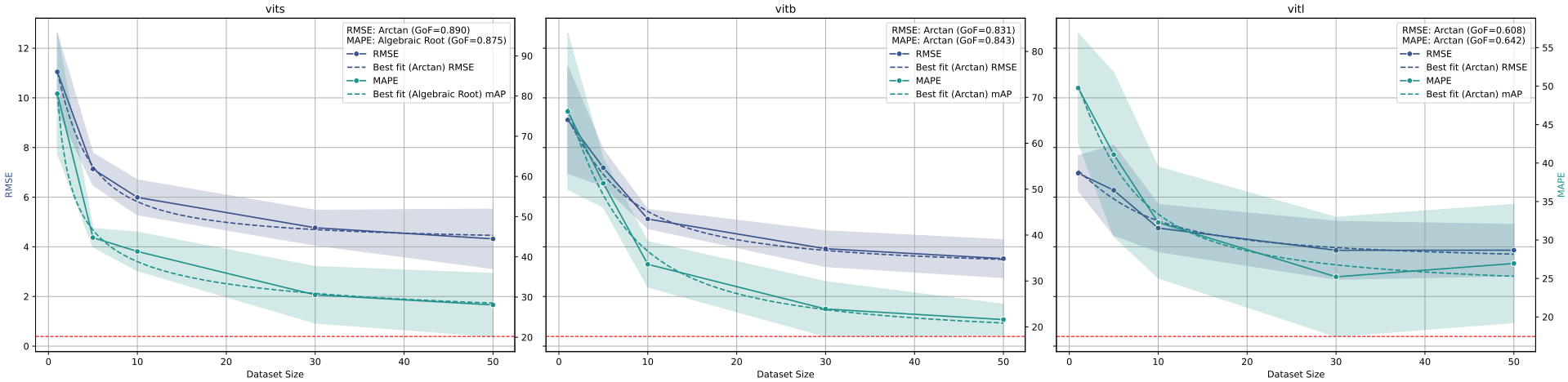


Figure 9: The figure shows the relationship between shots and model performance for the CD-ViTO model trained and tested on ID datasets. The x-axis represents the number of shots. The solid lines represent the mean values, while the dashed lines indicate the shots amount/metric prediction model. The left and right y-axis represents the $RMSE$ and MAPE values respectively. The red dashed horizontal line represents the benchmark $RMSE$ value of 0.39. The combined legend in the upper right corner of each subplot shows the goodness of fit (GoF) for both $RMSE$ and MAPE.

The best result achieved by the CD-ViTO model was a $RMSE$ of 3.9 with ViT-B backbone and 50 shots to build the prototypes, which is substantially worse than the benchmark value of 0.39 (10 times higher). This corresponds to a MAPE on counting of about 25and a mAP of about 0.5 (Figures 8 and 9). It corresponds roughly to a miscounted plant over four as it is visible looking to some predictions of this model in Figure 10. The models fitted on metrics show a reliable GoF for all the metrics, indicating that the model performance is highly predictable by the number of shots. These also show that any CD-ViTO size model would not achieve the benchmark with any shot amount, even if the number of shots were increased beyond those tested.

2.1.3.4 Zero-shots object detectors

Figure 11 shows the relationship between the zero-shots model settings and model performance tested on ID testing datasets. Not all the model settings were able to predict the whole testing dataset. For example, the owlv2-base-patch16-finetuned model was not able to generate any prediction with any prompt for any image of the ID testing. A dataset size relationship with metrics could not be established as zero-shots models do not require fine-tuning training data. None of the zero-shots model settings reached the benchmark. This is particularly true for the R^2 values, which were always below 0, indicating poor predictive performance. The $RMSE$ values ranged from approximately 5 to 25, significantly higher than those observed in the many-shots and few-shots models. Additionally, MAPE values were

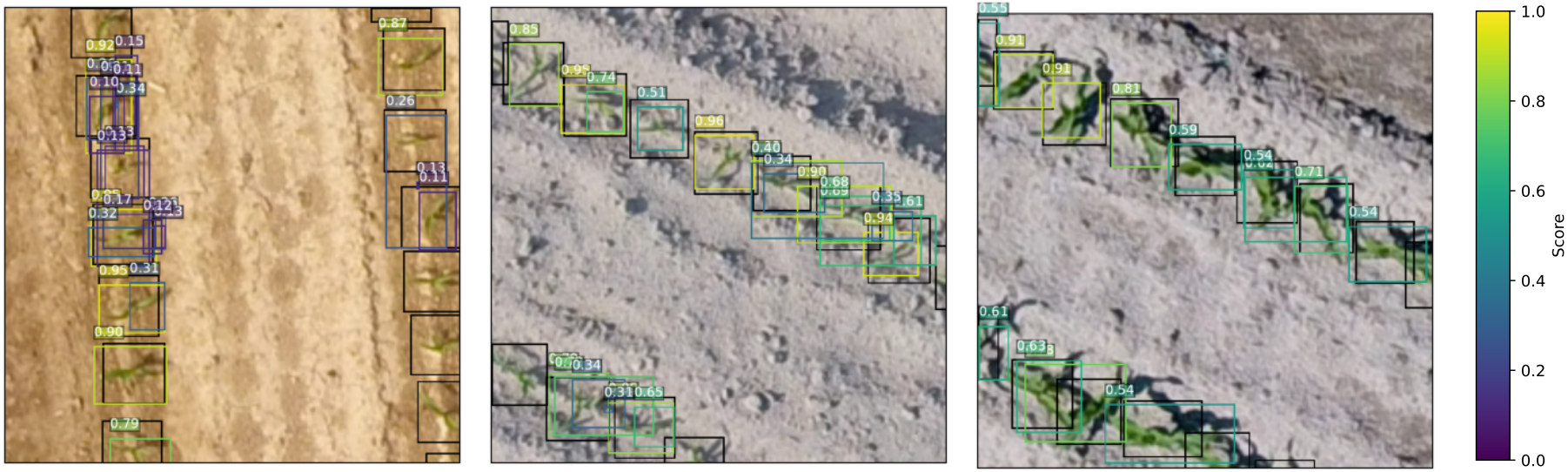


Figure 10: 50 shots CD-ViTO with ViT-B backbone predictions on the 1, 2 and 3 ID test datasets tile examples, respectively from the left hand side to the right. Black bounding boxes are the ground truth annotations, while the bounding boxes in viridis color scale are the model predictions.

also considerably elevated, ranging from around 40 to 140. Furthermore, the mAP values were lower than those of the many and few-shots models for all model settings, except for the owlv2-large-patch14-finetuned model, for which very few images were successfully predicted with an mAP comparable to that of the best few-shots model (50-shots ViT-B backbone). Some rare case of good predictions were even more accurate than the few-shots best performance, as shown in Figure 12.

2.1.4 Discussion

2.1.4.1 Dataset Source Impact on Object Detection Performance

Our experiments clearly demonstrate the critical importance of dataset source for successful arable crop seedling detection. None of the tested models, regardless of architecture or parameter amount, achieved the benchmark R^2 value of 0.85 when trained on out-of-distribution (OOD) datasets. Several inherent biases in our datasets likely influenced model performance. This aligns with previous findings by David et al. [22] and Andvaag et al. [13], who similarly reported significantly lower performance when using training samples from sources different from the inference dataset.

The domain gap challenge is particularly pronounced in agricultural applications, where environmental conditions, lighting, camera parameters, and plant growth stages vary substantially across datasets.

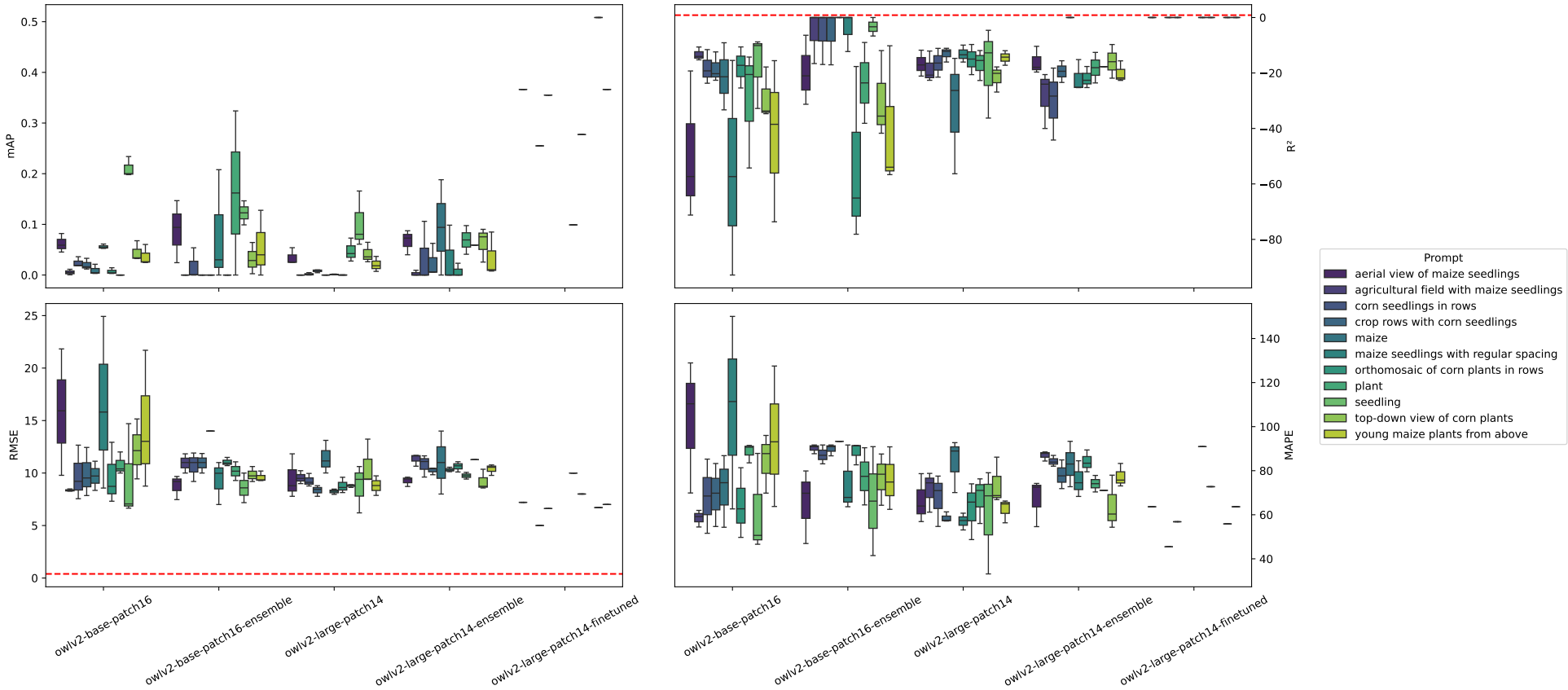


Figure 11: The figure shows the relationship between the OWLv2 model size, used prompt and model performance. The x-axis represents the model settings and the model size. Colors represent the different prompts used. The four subplots show the mAP (upper left corner), R^2 (upper right corner), $RMSE$ (lower left corner), and MAPE (lower right corner) values. The red dashed horizontal line in the R^2 and the $RMSE$ subplots represents respectively the benchmark of 0.85 and 0.39.

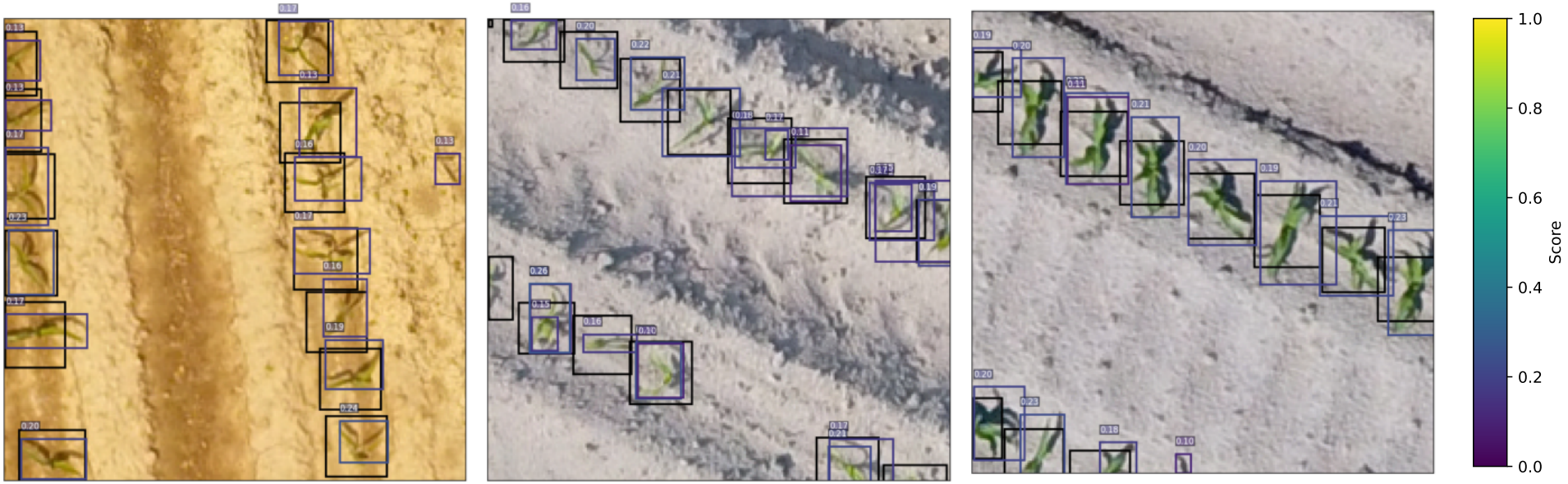


Figure 12: The best predictions with the OWLv2 model. The ID_1, ID_2 and ID_3 datasets respectively from the left hand side to the right. Prediction with owlv2-base-patch16-ensemble model of the ID_1 dataset, and with owlv2-base-patch16 model on the other two datasets. All the predictions are made with the prompt ”seedling”. Black bounding boxes are the ground truth annotations, while the bounding boxes in viridis color scale are the model predictions.

The failure of OOD training highlights that visual features learned from one orthomosaic do not generalize well to others without significant adaptation. As the goodness-of-fit (GoF) of the models explicating the relationship between dataset size and performance was always below 0.2, one can argue that the interval of dataset size tested was too narrow to achieve a good fit or that other variables play an important role in determining model performance. Both cases are likely to be true, but also the maximum OOD dataset size that was tested (1168) was really small in respect to other studies that use training datasets of tens of thousand of images to achieve such benchmarks [15]. This further highlights the importance of collecting in-domain training data, as the minimum OOD dataset size and quality to train an object detector to count arable crops seedling that generalizes to all the real-world cases is difficult even to establish with a limited dataset.

Despite the poor performance of OOD dataset trained models, some of them showed a low $MAPE$ value of less than 20%, not enough to consider the models for direct inferencing but rather as an annotation tool for the ID dataset.

2.1.4.2 Many-Shot Object Detection: Architecture and Dataset Requirements

Our results reveal important relationships between model architecture, count metrics, and minimum dataset requirements. Within YOLO-family models, we observed that the lightweight YOLOv5n, YOLOv5s, and YOLOv8n achieved the benchmark with 130, 130, and 110 sam-

ples respectively. As already well-known, increasing model complexity in CNN-based architectures corresponded to increased dataset size requirements. Conversely, for transformer-mixed models like RT-DETR, we observed different patterns, with RT-DETR L achieving the benchmark with only 60 samples while the larger RT-DETR X required 100 samples. The empirical models of dataset size versus performance showed comparable GoF between RT-DETR and YOLO-family models, except for YOLOv5n which showed a particularly high GoF. This suggests that transformer-based models have the same predictability to reach the benchmark with the reported dataset size as the CNN-based models, except for YOLOv5n which has a higher predictability to reach the benchmark given the same dataset size. Overall, transformer-based models may require fewer samples to achieve the same performance as CNN-based models, potentially due to their ability to capture long-range dependencies and contextual information more effectively. A visible side-effect of the adoption of transformer-mixed models is the higher computational cost of the training phase in terms of time and memory, that could be a limitation for some applications. This creates a practical tradeoff for practitioners: whether to use a simpler CNN-based model like YOLOv5n and invest in collecting more annotated images (approximately 130), or to allocate more computational resources for a transformer-mixed model like RT-DETR L that can achieve comparable performance with roughly half the amount of labeled data (approximately 60 images).

The predictability of model performance based on dataset size (as evidenced by GoF values exceeding 0.3 for successful models) pro-

vides practical guidance for practitioners. The relationship between dataset size and performance (modeled using logarithmic, arctangent, or algebraic root functions, depending on best fit) suggests diminishing returns beyond certain thresholds, which can help inform efficient resource allocation for annotation efforts.

2.1.4.3 Dataset Quality Trade-offs

Our investigation into minimum dataset quality requirements revealed that models can tolerate some reduction in annotation quality while still maintaining benchmark performance achieved with the same training dataset size. YOLOv5n, YOLOv5s, and YOLOv8n achieved the benchmark with 85%, 90%, and 85% of the original dataset quality, while RT-DETR X required only 65%. Notably, RT-DETR L failed to maintain benchmark performance with any reduction in annotation quality, suggesting different sensitivity to annotation errors.

This difference in quality tolerance between RT-DETR L and RT-DETR X can be explained by considering their respective minimum dataset sizes. RT-DETR L was tested with quality reductions on its minimum benchmark-achieving dataset size of just 60 samples, while RT-DETR X was tested with 100 samples. With fewer training examples, RT-DETR L becomes more sensitive to the quality of each individual annotation, as each annotation represents a larger proportion of the total learning signal. In contrast, RT-DETR X, with its larger training dataset, can better compensate for quality reductions by leveraging redundancy across more examples.

These findings provide valuable insights for practical applications, as they suggest that in some cases, it may be more efficient to collect a larger quantity of moderately-quality annotations rather than focusing on perfect annotations for a smaller dataset. This also indicates potential for semi-automated annotation workflows, where machine assistance in annotation (which may introduce some errors) could be acceptable for many applications.

2.1.4.4 Few-Shot and Zero-Shot Approaches: Current Limitations

Despite recent advances in few-shot and zero-shot learning, our experiments reveal significant limitations in these approaches for precise maize seedling detection. The best CD-ViTO few-shot model achieved a $RMSE$ of 3.9 with 50 shots (using ViT-B backbone), substantially below the benchmark requirement of 0.39. Similarly, zero-shot models like OWLv2 performed poorly regardless of prompt engineering efforts.

These results contrast with the promising performance reported for few-shot and zero-shot methods in general object detection benchmarks [79, 62]. Several factors may explain this gap: First, the domain-specific nature of aerial maize seedling imagery, where subtle textural differences and high intra-class variability are prevalent, severely challenges models pre-trained on general object detection datasets. As illustrated in the few-shot experiments (Figure 8), increasing the number of shots leads to nonlinear improvements in metrics such as R^2 and mAP (following an arctan-like trend), yet

the error metric ($RMSE$) remain significantly above the benchmark. This saturation effect suggests that even with more than usual maximum tested prototypes (50 instead of 30), the models struggle to capture the fine-grained visual cues necessary for precise seedling detection. Moreover, the zero-shot results (Figure 11) reveal a pronounced sensitivity to prompt phrasing, with all variants, including ensemble and fine-tuned versions of OWLv2, consistently failing to approach acceptable error rates. These observations imply that both the inherent complexity of the task and the limitations of current few-shot and zero-shot frameworks necessitate more domain-specific strategies. Addressing these challenges through domain-specific adaptations could help narrow the performance gap, potentially making few-shot and zero-shot methods more competitive for arable crop seedling detection. Interestingly, the few-shot and the zero-shot models were able to detect all the seedlings without false positives in few cases. It would be interesting to investigate the possible ways to retain these images and use them to populate the training dataset for a many-shots model.

2.1.4.5 Handcrafted Methods in the Deep Learning Era

Despite the focus on deep learning approaches, our handcrafted (HC) object detector demonstrated strong performance on the testing datasets (R^2 from 0.87 to 0.95). However, a significant limitation was the small proportion of tiles (1.8% to 7.8%) for which it could provide reliable annotations. This illustrates the classic trade-off of

rule-based systems: high precision in constrained scenarios but limited generalizability.

These findings suggest that HC methods may still have value in a hybrid approach, where they provide high-quality annotations on a subset of data, which can then be used to bootstrap deep learning models. Such an approach could be particularly valuable for specialized agricultural applications where annotation resources are limited.

This approach is highly adopted in industry, where the HC method is used to annotate the training dataset and the deep learning model is used to predict the real-world cases, but it introduces a possible bias in the training dataset that could be a limitation for the model generalization. The main problem is that the HC1 method relies on color thresholding that filter the objects based on the color of the objects. That could be not the best way to annotate the training dataset for a deep learning model that could learn more complex features of the objects, but also the ones selected by the HC1 method.

2.1.4.6 Implications for Practical Applications

Our study has several practical implications for developing arable crop seedling detection systems. First, collecting in-domain training data is non-negotiable for achieving benchmark performance. Finding a way to automatically obtain the training dataset from the same distribution as the intended inference target is a key step in developing a robust object detector for arable crop seedling detection.

The logarithmic relationship between dataset size and performance suggests that initial annotation efforts should focus on reaching the minimum viable dataset size (60-130 images depending on architecture), after which additional annotations yield diminishing returns. This finding helps organizations optimize resource allocation for annotation efforts.

Our results also demonstrate that some reduction in annotation quality is acceptable, with models maintaining benchmark performance with 65-90% of the original quality. This suggests that semi-automated annotation workflows could be efficiently implemented for agricultural applications, potentially reducing the time and cost associated with manual annotation.

Current few-shot and zero-shot methods, while promising, are not yet viable replacements for traditional object detection approaches in seedling detection or counting tasks. However, they might still serve auxiliary roles in the annotation pipeline.

Hybrid approaches combining handcrafted methods with deep learning models could provide a practical solution for achieving benchmark performance. We observed that OOD many-shots, few-shots, and zero-shots models are occasionally able to produce annotations with sufficient quality for training ID many-shots models. A promising direction for future work would be to develop methods for automatically identifying and leveraging these high-quality annotations. Specifically, the HC2 component of our handcrafted approach could potentially be used to filter and validate annotations produced by these models, overcoming the color-thresholding bias introduced by

HC1 while maintaining the agronomic knowledge encoded in HC2's row-pattern validation.

2.1.4.7 Future Work

In this study, we focused on the minimum dataset requirements for fine-tuning pre-trained models for the downstream task of counting arable crop seedlings through object detection. We did not explore the potential benefits of using domain-specific backbones. Future work could investigate whether dataset size requirements could be further reduced by using backbones pre-trained on agricultural imagery, particularly aerial orthomosaics of crop fields. Such domain-specific pre-training might allow models to learn more relevant features for crop detection tasks, potentially reducing the amount of in-domain data needed for fine-tuning.

2.1.5 Conclusions

This study demonstrates that successful maize seedling detection requires in-domain training data, with out-of-distribution training requiring unreasonable dataset size to achieve benchmark performance across all tested models. We established minimum dataset requirements for several architectures, finding that lightweight YOLO models achieve benchmark performance with 110-130 samples, while certain transformer-mixed models like RT-DETR require as few as 60 samples. Models showed varying tolerance for reduced annotation quality, with some maintaining performance with only 65-90%

of original annotation quality.

Despite advances in machine learning, neither few-shot nor zero-shot approaches currently meet precision requirements for arable crop seedling detection. Our handcrafted algorithm achieved excellent performance within its constraints, suggesting potential value in hybrid approaches combining rule-based methods with deep learning. These findings provide practical guidance for developing maize seedling detection systems, and possible ways to overcome the limitations of the current deep learning models for this application.

Bibliography

[1] [1409.0575] ImageNet Large Scale Visual Recognition Challenge.

[2] Estimates of Maize Plant Density from UAV RGB Images Using Faster-RCNN Detection Model: Impact of the Spatial Resolution.

[3] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks | IEEE Journals & Magazine | IEEE Xplore.

[4] Maize-seedling-detection Dataset $>$ Overview.

[5] Maize_seeding Dataset $>$ Overview.

[6] Meta-Learning-Based Incremental Few-Shot Object Detection | IEEE Journals & Magazine | IEEE Xplore.

[7] You Only Look Once: Unified, Real-Time Object Detection | IEEE Conference Publication | IEEE Xplore.

[8] PP 1/333 (1) Adoption of digital technology for data generation for the efficacy evaluation of plant protection products. *EPPO Bulletin*, page epp.13037, November 2024.

[9] Fatih Cagatay Akyon, Sinan Onur Altinuc, and Alptekin Temizel. Slicing Aided Hyper Inference and Fine-tuning for Small Object Detection. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 966–970, October 2022.

[10] Khaled Alhazmi, Walaa Alsumari, Indrek Seppo, Lara Podkuiko, and Martin Simon. Effects of annotation quality on model performance. In *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 063–067, April 2021.

[11] Hamed Amini Amirkolaee, Miaojing Shi, Lianghua He, and Mark Mulligan. AdaTreeFormer: Few shot domain adaptation for tree counting from a single high-resolution image. *ISPRS Journal of Photogrammetry and Remote Sensing*, 214:193–208, August 2024.

[12] Ayoub Benali Amjoud and Mustapha Amrouch. Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review. *IEEE access : practical innovations, open solutions*, 11:35479–35516, 2023.

[13] Erik Andvaag, Kaylie Krys, Steven J. Shirtliffe, and Ian Stavness. Counting Canola: Toward Generalizable Aerial Plant Detection Models. *Plant phenomics (Washington, D.C.)*, 6:0268, November 2024.

[14] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data Augmentation Generative Adversarial Networks, March 2018.

[15] 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, August 2024.

[16] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-Shot Object Detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 384–400, 2018.

[17] Abel Barreto, Philipp Lottes, Facundo Ramón Ispizua Yamati, Stephen Baumgarten, Nina Anastasia Wolf, Cyrill Stachniss, Anne-Katrin Mahlein, and Stefan Paulus. Automatic UAV-based counting of seedlings in sugar-beet field and extension to maize and strawberry. *Computers and Electronics in Agriculture*, 191:106493, December 2021.

[18] L. Brigato and L. Iocchi. A Close Look at Deep Learning with Small Data, October 2020.

[19] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers, May 2020.

[20] Clément Chadebec and Stéphanie Allassonnière. Data Augmentation with Variational Autoencoders and Manifold Sampling. In Sandy Engelhardt, Ilkay Oksuz, Dajiang Zhu, Yixuan Yuan, Anirban Mukhopadhyay, Nicholas Heller, Sharon Xiaolei Huang, Hien Nguyen, Raphael Sznitman, and Yuan Xue, editors, *Deep Generative Models, and Data Augmentation,*

Labelling, and Imperfections, pages 184–192, Cham, 2021. Springer International Publishing.

[21] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning Augmentation Policies from Data, April 2019.

[22] Etienne David, Gaëtan Daubige, François Joudelat, Philippe Burger, Alexis Comar, Benoit de Solan, and Frédéric Baret. Plant detection and counting from high-resolution RGB images acquired from UAVs: Comparison between deep-learning and handcrafted methods with application to maize, sugar beet, and sunflowe, 2021.

[23] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale, June 2021.

[24] Xianzhi Du, Tsung-Yi Lin, Pengchong Jin, Golnaz Ghiasi, Mingxing Tan, Yin Cui, Quoc V. Le, and Xiaodan Song. SpineNet: Learning Scale-Permuted Backbone for Recognition and Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11592–11601, 2020.

[25] FAO. In *Agricultural Production Statistics 2010–2023*, volume Analytical Briefs. FAOSTAT, Rome, 2024.

[26] Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. In Martin A. Fischler and Oscar Firschein, editors, *Readings in Computer Vision*, pages 726–740. Morgan Kaufmann, San Francisco (CA), January 1987.

[27] Kun Fu, Tengfei Zhang, Yue Zhang, Menglong Yan, Zhonghan Chang, Zhengyuan Zhang, and Xian Sun. Meta-SSD: Towards Fast Adaptation for Few-Shot Object Detection With Meta-Learning. *IEEE access : practical innovations, open solutions*, 7:77597–77606, 2019.

[28] Yuqian Fu, Yu Wang, Yixuan Pan, Lian Huai, Xingyu Qiu, Zeyu Shangguan, Tong Liu, Yanwei Fu, Luc Van Gool, and Xingqun Jiang. Cross-Domain Few-Shot Object Detection via Enhanced Open-Set Object Detector, September 2024.

[29] Héctor García-Martínez, Héctor Flores-Magdaleno, Abdul Khalil-Gardezi, Roberto Ascencio-Hernández, Leonardo Tijerina-Chávez, Mario A. Vázquez-Peña, and Oscar R. Mancilla-Villa. Digital Count of Corn Plants Using Images Taken by Unmanned Aerial Vehicles and Cross Correlation of Templates. *Agronomy*, 10(4):469, April 2020.

[30] Tingting Geng, Haiyang Yu, Xinru Yuan, Ruopu Ma, and Pengao Li. Research on Segmentation Method of Maize Seedling Plant Instances Based on UAV Multispectral Remote Sensing Images. *Plants*, 13(13):1842, January 2024.

[31] Micah Goldblum, Hossein Souri, Renkun Ni, Manli Shu, Viraj Prabhu, Gowthami Somepalli, Prithvijit Chattopadhyay, Mark Ibrahim, Adrien Bardes, Judy Hoffman, Rama Chellappa, Andrew Gordon Wilson, and Tom Goldstein. Battle of the Backbones: A Large-Scale Comparison of Pretrained Models across Computer Vision Tasks, November 2023.

[32] Nico Heider, Lorenz Gunreben, Sebastian Zürner, and Martin Schieck. A Survey of Datasets for Computer Vision in Agriculture: A catalogue of high-quality RGB image datasets of natural field scenes. *45. GIL-Jahrestagung, Digitale Infrastrukturen für eine nachhaltige Land-, Forst und Ernährungswirtschaft*, pages 35–47, 2025.

[33] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization, June 2020.

[34] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep Learning Scaling is Predictable, Empirically, December 2017.

[35] Gabriel Huang, Issam Laradji, David Vazquez, Simon Lacoste-Julien, and Pau Rodriguez. A Survey of Self-Supervised and Few-Shot Object Detection, August 2022.

[36] Pranav Jeevan and Amit Sethi. Which Backbone to Use: A

Resource-efficient Domain Specific Comparison for Computer Vision, June 2024.

[37] Yu Jiang, Changying Li, Andrew H. Paterson, and Jon S. Robertson. DeepSeedling: Deep convolutional network and Kalman filter for plant seedling detection and counting in the field. *Plant Methods*, 15(1):141, November 2019.

[38] Longlong Jing and Yingli Tian. Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey, February 2019.

[39] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. GitHub Ultralytics YOLO, January 2023.

[40] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-Shot Object Detection via Feature Reweighting, October 2019.

[41] Azam Karami, Melba Crawford, and Edward J. Delp. Automatic Plant Counting and Location Based on a Few-Shot Learning Technique. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:5872–5886, 2020.

[42] Sushma Katari, Sandeep Venkatesh, Christopher Stewart, and Sami Khanal. Integrating Automated Labeling Framework for Enhancing Deep Learning Models to Count Corn Plants Using UAS Imagery. *Sensors*, 24(19):6467, January 2024.

[43] Asifullah Khan, Zunaira Rauf, Anabia Sohail, Abdul Rehman Khan, Hifsa Asif, Aqsa Asif, and Umair Farooq. A survey of

the vision transformers and their CNN-transformer based variants. *Artificial Intelligence Review*, 56(3):2917–2970, December 2023.

[44] Rahima Khanam and Muhammad Hussain. YOLOv11: An Overview of the Key Architectural Enhancements, October 2024.

[45] Bruno T. Kitano, Caio C. T. Mendes, André R. Geus, Henrique C. Oliveira, and Jefferson R. Souza. Corn Plant Counting Using Deep Learning and UAV Images. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2019.

[46] Karl Kraus. *Photogrammetry: Geometry from Images and Laser Scans*. De Gruyter, October 2011.

[47] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[48] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[49] Wenwen Li and Yun Zhang. DC-YOLO: An improved field plant detection algorithm based on YOLOv7-tiny. *Scientific Reports*, 14(1):26430, November 2024.

[50] Yang Li, Zhiyuan Bao, and Jiangtao Qi. Seedling maize counting method in complex backgrounds based on YOLOV5 and

Kalman filter tracking algorithm. *Frontiers in Plant Science*, 13, November 2022.

[51] Yong Li, Naipeng Miao, Liangdi Ma, Feng Shuang, and Xingwen Huang. Transformer for object detection: Review and benchmark. *Engineering Applications of Artificial Intelligence*, 126:107021, November 2023.

[52] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-SGD: Learning to Learn Quickly for Few-Shot Learning, September 2017.

[53] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection, February 2018.

[54] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context, February 2015.

[55] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with Grounded Pre-training for Open-Set Object Detection. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 38–55, Cham, 2025. Springer Nature Switzerland.

[56] Shuaibing Liu, Dameng Yin, Haikuan Feng, Zhenhai Li, Xiaobin Xu, Lei Shi, and Xiuliang Jin. Estimating maize seedling

number with UAV RGB images and advanced image processing methods. *Precision Agriculture*, 23(5):1604–1632, October 2022.

[57] Wenxin Liu, Jing Zhou, Biwen Wang, Martin Costa, Shawn M. Kaeppler, and Zhou Zhang. IntegrateNet: A Deep Learning Network for Maize Stand Counting From UAV Imagery by Integrating Density and Local Count Maps. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.

[58] Hao Lu, Zhiguo Cao, Yang Xiao, Bohan Zhuang, and Chunhua Shen. TasselNet: Counting maize tassels in the wild via local counts regression network. *Plant Methods*, 13(1):79, November 2017.

[59] Mélissande Machefer, François Lemarchand, Virginie Bonnefond, Alasdair Hitchins, and Panagiotis Sidiropoulos. Mask R-CNN Refitting Strategy for Plant Counting and Sizing in UAV Imagery. *Remote Sensing*, 12(18):3015, January 2020.

[60] Rafid Mahmood, James Lucas, David Acuna, Daiqing Li, Jonah Philion, Jose M. Alvarez, Zhiding Yu, Sania Fidler, and Marc T. Law. How Much More Data Do I Need? Estimating Requirements for Downstream Tasks. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 275–284, New Orleans, LA, USA, June 2022. IEEE.

[61] Uwe Meier, Hermann Bleiholder, Liselotte Buhr, Carmen Feller, Helmut Hack, Martin Heß, Peter D. Lancashire, Uta Schnock,

Reinhold Stauß, Theo van den Boom, Elfriede Weber, and Peter Zwerger. The BBCH system to coding the phenological growth stages of plants – history and publications –. *Journal für Kulturpflanzen*, 61(2):41–52, February 2009.

[62] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling Open-Vocabulary Object Detection. *Advances in Neural Information Processing Systems*, 36:72983–73007, December 2023.

[63] Samuel G. Müller and Frank Hutter. TrivialAugment: Tuning-free Yet State-of-the-Art Data Augmentation, August 2021.

[64] Nhat-Duy Nguyen, Tien Do, Thanh Duc Ngo, Duy-Dinh Le, and Cesare F. Valenti. An Evaluation of Deep Learning Methods for Small Object Detection. *JECE*, 2020, January 2020.

[65] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision, February 2024.

[66] Aref Miri Rekavandi, Shima Rashidi, Farid Boussaid, Stephen Hoefs, Emre Akbas, and Mohammed bennamoun. Transform-

ers in Small Object Detection: A Benchmark and Survey of State-of-the-Art, September 2023.

[67] Gianmarco Roggiolani, Federico Magistri, Tiziano Guadagnino, Jan Weyler, Giorgio Grisetti, Cyrill Stachniss, and Jens Behley. On Domain-Specific Pre- Training for Effective Semantic Perception in Agricultural Robotics. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11786–11793, May 2023.

[68] Liangbing Sa, Chongchong Yu, Zhaorui Hong, Tong Zheng, and Sihan Liu. A broader study of cross-domain few-shot object detection. *Applied Intelligence*, 53(23):29465–29485, December 2023.

[69] Connor Shorten and Taghi M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1):60, July 2019.

[70] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era, August 2017.

[71] Mingxing Tan, Ruoming Pang, and Quoc V. Le. EfficientDet: Scalable and Efficient Object Detection, July 2020.

[72] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Process-*

ing Systems, NIPS'17, pages 6000–6010, Red Hook, NY, USA, December 2017. Curran Associates Inc.

[73] K. Velumani, R. Lopez-Lozano, S. Madec, W. Guo, J. Gillet, A. Comar, and F. Baret. Estimates of Maize Plant Density from UAV RGB Images Using Faster-RCNN Detection Model: Impact of the Spatial Resolution. *Plant phenomics (Washington, D.C.)*, 2021:9824843, January 2021.

[74] Biwen Wang, Jing Zhou, Martin Costa, Shawn M. Kaeppler, and Zhou Zhang. Plot-Level Maize Early Stage Stand Counting and Spacing Detection Using Advanced Deep Learning Algorithms Based on UAV Imagery. *Agronomy*, 13(7):1728, July 2023.

[75] Dongxue Wang, Rajamohan Parthasarathy, and Xian Pan. Advancing Image Recognition: Towards Lightweight Few-shot Learning Model for Maize Seedling Detection. In *Proceedings of the 2024 International Conference on Smart City and Information System*, ICSCIS '24, pages 635–639, New York, NY, USA, August 2024. Association for Computing Machinery.

[76] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language*

Processing: System Demonstrations, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[77] Kesheng Wu, Ekow Otoo, and Arie Shoshani. Optimizing connected component labeling algorithms. January 2005.

[78] Xiongwei Wu, Doyen Sahoo, and Steven Hoi. Meta-RCNN: Meta Learning for Few-Shot Object Detection. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, pages 1679–1687, New York, NY, USA, October 2020. Association for Computing Machinery.

[79] Guanyu Xu, Zhiwei Hao, Yong Luo, Han Hu, Jianping An, and Shiwen Mao. DeViT: Decomposing Vision Transformers for Collaborative Inference in Edge Devices, September 2023.

[80] Gongjie Zhang, Zhipeng Luo, Kaiwen Cui, and Shijian Lu. *Meta-DETR: Few-Shot Object Detection via Unified Image-Level Meta-Learning*. March 2021.

[81] Song Zhang, Yehua Yang, Lei Tu, Tianling Fu, Shenxi Chen, Fulang Cen, Sanwei Yang, Quanzhi Zhao, Zhenran Gao, and Tengbing He. Comparison of YOLO-based sorghum spike identification detection models and monitoring at the flowering stage. *Plant Methods*, 21(1):20, February 2025.

[82] Kai Zhao, Lulu Zhao, Yanan Zhao, and Hanbing Deng. Study on Lightweight Model of Maize Seedling Object Detection Based on YOLOv7. *Applied Sciences*, 13(13):7731, January 2023.

[83] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. DETRs Beat YOLOs on Real-time Object Detection, April 2024.

[84] Zhuofan Zong, Guanglu Song, and Yu Liu. DETRs with Collaborative Hybrid Assignments Training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6725–6735, Paris, France, October 2023. IEEE.

[85] Hongwei Zou, Hao Lu, Yanan Li, Liang Liu, and Zhiguo Cao. Maize tassels detection: A benchmark of the state of the art. *Plant Methods*, 16(1):108, August 2020.