This ICCV workshop paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Adapting Vision Foundation Models for Plant Phenotyping

Feng Chen[1], Mario Valerio Giuffrida[2], Sotirios A. Tsaftaris[1]

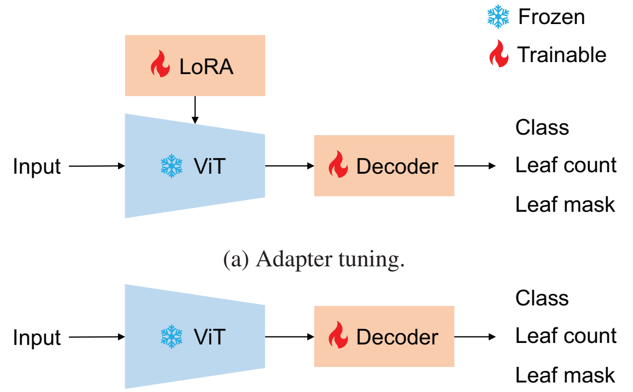
[1]The University of Edinburgh, U.K.
[2]The University of Nottingham, U.K.
{feng.chen, s.tsaftaris}@ed.ac.uk
valerio.giuffrida@nottingham.ac.uk

Abstract

Foundation models are large models pre-trained on tremendous amount of data. They can be typically adapted to diverse downstream tasks with minimal effort. However, as foundation models are usually pre-trained on images or texts sourced from the Internet, their performance in specialized domains, such as plant phenotyping, comes into question. In addition, fully fine-tuning foundation models is time-consuming and requires high computational power. This paper investigates the efficient adaptation of foundation models for plant phenotyping settings and tasks. We perform extensive experiments on fine-tuning three foundation models, MAE, DINO, and DINOv2 on three essential plant phenotyping tasks: leaf counting, instance segmentation, and disease classification. In particular, the pre-trained backbones are kept frozen, while two distinct fine-tuning methods are evaluated, namely adapter tuning (using LoRA) and decoder tuning. The experimental results show that a foundation model can be efficiently adapted to multiple plant phenotyping tasks, yielding similar performance as the state-of-the-art (SoTA) models specifically designed or trained for each task. Despite exhibiting great transferability over different tasks, the fine-tuned foundation models perform slightly worse than the SoTA task-specific models in some scenarios, which requires further investigation.

1. Introduction

Plant phenotyping aims to quantitatively evaluate structural and functional attributes of plants, which are crucial across several domains in modern agriculture, such as crop improvement, disease and pest management, and climate resilience [50]. In the past decade, deep learning has become an important tool in different plant phenotyping tasks, such as leaf counting and segmentation [14, 17, 20], root segmentation [44, 49], wheat spikelet localization [7, 40], and disease classification [41, 42].



(a) Adapter tuning.

(b) Decoder tuning.

Figure 1: Adapter and decoder tuning. We employ a pre-trained frozen ViT backbone (MAE, DINO, or DINOv2) and a task specific decoder (linear layer for classification and counting, Feature Pyramid Network (FPN) + Mask RCNN decoder for instance segmentation). In adapter tuning, the parameters added by LoRA and the decoder are both trainable, while in decoder tuning, only the parameters associated with the decoder is trainable.

However, the majority of these models are characterised by a task-specific architectural design: while they are proficient at performing the tasks for which they were designed or trained, their performance decreases when applied to other tasks. Such lack of versatility makes the implementation (and deployment) of deep learning models inefficient, as new models need to be devised, developed, and trained for each specific task. Recent development of foundational models [4] is a promising avenue to solve this problem for the plant phenotyping community.

Foundation models are models with substantial number of trainable parameters and pre-trained on an extensive and diverse set of data, which have the potential to easily adapt to a wide range of unseen tasks [4]. One famous example of foundation models is ChatGPT [5], a large language foundation model pre-trained on a vast amount of internet data using autoregressive language modelling (that is, predicting the next word in a sentence given all of the previous words). ChatGPT has shown remarkable versatility in performing novel tasks that it is not trained for, such as programming and translation. In computer vision, a recent example of foundation models is the Segment Anything Model (SAM) [31]. SAM is pre-trained on a huge dataset called SA-1B, comprising 11 million images supplemented with 1 billion mask labels, to perform prompted image segmentation tasks. These two recent examples showcase the flexibility and wide-ranging applicability of foundation models, spanning from natural language processing to computer vision.

Despite the benefits of foundation models, these models have two major constraints. First, foundation models are typically trained on general domain data, lacking specialized and domain-specific knowledge. This limitation becomes noticeable when foundation models are applied to areas that differ from general context and applications, such as medical imaging, industrial inspection and agriculture [24, 28, 29]. Fine-tuning foundation models on new data and tasks is promising to ease the effect of domain shift. However, this leads to a second limitation: foundation models are huge (in terms of parameters) and complicated. As they are characterized by billions of trainable parameters, fine-tuning such large models is time-consuming and requires a hardware infrastructure that is prohibitive to the majority of users in specialized domains.

Conversely, tuning only the task-specific decoder, such as the linear layers for image classification, while freezing the rest of the model, makes the transfer of large models to novel tasks more feasible. However, decoder tuning may lead to sub-optimal performance [16, 19]. In recent years, there has been growing interest in the concept of parameter-efficient fine-tuning (PEFT) as a solution to achieve the balance between tuning costs and performance [34]. Different from decoder tuning, PEFT fine-tunes a small subset of the parameters in the model backbone, while the majority of parameters in the backbone remain unchanged [13]. Adapters, important members of the PEFT family, are newly integrated trainable modules that fit between layers of a pre-trained model [25]. These modules are powerful and efficient, delivering comparable performance to full fine-tuning while significantly lowering computational costs [9, 10, 27]. Thus, adapters provide a middle ground to efficiently fine-tune foundation models to unseen scenarios and tasks, while preserving their flexibility.

To alleviate the tedium and expense associated with developing task-specific models individually, the key objective of this study is to answer the question: *Is it possible to tackle diverse plant phenotyping tasks by efficiently fine-tuning pre-trained foundation models?* To answer, we focus on three leading foundation models based on vision transformer (ViT) [15]: Masked Autoencoders (MAE) [21], DINO [6] and DINOv2 [38]. We perform extensive experiments to evaluate the performance of fine-tuning these models using decoder tuning and a popular adapter, the Low-Rank Adaptation (LoRA) [27], on the following plant phenotyping-related tasks: leaf counting, instance segmentation, and disease classification. Figure 1 summarizes the methodology of this study. The datasets used in our experiments are diverse and collected from both controlled and in-field environments, which allows us to explore the effectiveness and adaptability of foundation models under various conditions. The main contributions of this work are:

1. To the best of our knowledge, this study is the first to investigate the adaptation of foundation models (using LoRA) on diverse plant phenotyping tasks. Our experimental work contributes to a critical evaluation of these advanced methodologies, as well as establishes valuable benchmarks in this field.

2. We also examine the circumstances under which foundation models are a better fit for the considered plant phenotyping applications. By also identifying and understanding limitations, our research provides crucial insights that could inform and guide future developments of foundation models for plant phenotyping.

2. Related Work

This section reviews recent development of vision foundation models and adapters. Below we mainly cover self-supervised pre-trained foundation models as all the evaluated models in this work belong to this class.

2.1. Vision Foundation Models

Foundation models are characterized by a large number of trainable parameters and pre-trained with a big and diversified dataset [4]. In contrast to previous generations of AI models that focused on addressing individual tasks sequentially, foundation models possess the ability to adapt to a wide range of downstream tasks in various domains.

Building upon the success of large language models [5, 12], the design of vision foundation models often leverage Vision Transformers (ViTs) [15] and self-supervised pre-training techniques to excel in visual tasks. ViT operates by dividing an input image into patches and processes them as a sequence of embedded vectors, thereby enabling the model to identify and handle long-range correlations throughout the whole image. Self-supervised learning is a machine learning technique where models are trained to learn representations from unlabeled data [36]. This is done by creating learning tasks from the training data themselves, *e.g*. predicting a portion of an image given the rest of the image. Self-supervised learning allows models to be pre-trained on large-scale unlabeled data and to gain great transfer ability to diverse downstream tasks. Below we briefly introduce the mechanism of several ViT-based self-supervised pre-trained foundation models.

Masked Autoencoder (MAE) [21]. MAE randomly masks patches of an input image and trains an asymmetric encoder-decoder model to reconstruct the original image. The unique characteristic of the encoder is that it focuses solely on encoding the unmasked patches into latent vectors. In contrast, the decoder uses a combination of masked tokens and encoded unmasked tokens to recreate the original image. This approach has recently been proven successful in the Segment Anything Model (SAM) [31], gaining significant recognition in computer vision.

DINO [6]. DINO leverages self-distillation to train two ViTs, referred to as the teacher and student models. These models are structurally identical but with distinct parameters. The input images to each model are perturbed with random transformations, and DINO encourages the consistency between the outputs of the two models, promoting the learning of richer and invariant features. The latest version, DINOv2 [38], further proposes a new pipeline for curating large-scale datasets conducive to pre-training and distilling more complex models. This pipeline improves the applicability and scalability of DINO in diverse real-world settings.

BEiT [2]. Similar to BERT [12] used in natural language process, BEiT begins its training process by tokenizing the input images using a discrete variational autoencoder (dVAE). It then masks random sections of the image, and a ViT is trained to predict the original visual tokens corresponding to the masked image patches. To enhance this methodology, the subsequent version, BEiTv2 [39], implements vector-quantized knowledge distillation for training the tokenizer. This modification enables the tokenizer to discretize a continuous semantic space, expanding beyond the scope of low-level image pixels. The next advancement, BEiTv3 [46], includes the integration of language, vision, and multi-modal pre-training by unifying masked modeling across images, texts, and image-text pairs, reflecting a significant stride in multi-modal learning.

2.2. Adapters

Adapters are newly introduced trainable components that are inserted between the layers of a pre-trained model [25]. Adapters consist of a small number of parameters, aiming to efficiently adapt large models to diverse downstream tasks with the pre-trained weights frozen. Recent works [47, 16] demonstrate that adapters are not only compatible with a wide range of tasks, but also capable of delivering performance on par with (and sometimes surpassing) full fine-tuning. Below we provide a brief overview of several adapters that can be used on ViTs.

LoRA [27]. The hypothesis of the Low-Rank Adaptation (LoRA) is that the weight modifications during model adaptation exhibit a low intrinsic rank. Hence, the authors of LoRA propose to add pairs of trainable rank-decomposition weight matrices to each layer of the Transformer structure.

AdaptFormer [9]. AdaptFormer introduces lightweight side branches to the original MLP layers of ViTs. Within this new branch, the features are sequentially passed to down projection, ReLU activation and up projection. The output of this branch are then connected back to that of the original MLP layer using a residual connection.

ViT-Adapter [10]. ViT-Adapter is designed to effectively adapt plain ViTs to dense prediction tasks such as object detection and instance segmentation. To achieve this, three new components are introduced, spatial prior modules, spatial feature injectors, and multi-scale feature extractors, to effectively integrate image-related and task-related inductive biases into ViTs when adapting to downstream tasks.

The rapid development of foundation models and adapters have benefited various research domains. However, the research of these advanced approaches and their applications are still limited in plant phenotyping, which motivates this work. The evaluated foundation models in this paper include MAE, DINO and DINOv2 as MAE and DINO are pre-trained on the same dataset (ImageNet-1k [11]) with different training methods, while DINO and DINOv2 are pre-trained on different datasets (DINOv2 is pre-trained on a curated and larger dataset, LVD-142M [38]) using the same series of approach. LoRA, one of the most popular adapters in both natural language processing and computer vision, is adopted for adapter tuning.

3. Methodology

We investigate the potential of adapting foundation models to address multiple plant phenotyping tasks. To explore this, extensive experiments are performed on three fundamental plant phenotyping tasks: leaf counting, segmentation, and disease classification. Decoder tuning and LoRA are both investigated on three representative large-scale pre-trained ViTs: MAE, DINO, and DINOv2. We now proceed to detail the evaluated datasets and experimental setup.

3.1. Datasets

Leaf Counting and Segmentation. The CVPPP 2017 Leaf Counting and Segmentation Challenge Dataset (denoted as LCC and LSC below) [3, 37, 43] consists of RGB images of plants captured in controlled environment, and spans four domains denoted as A1, A2, A3, and A4. Domains A1, A2, and A4 contain different mutants of Arabidopsis plants, while domain A3 is composed of tobacco plants. In addition to A1 to A4, the test set of this dataset contains an extra domain, A5, where the images are a mixture, sampled from the four preceding domains. The images of this dataset are labeled with leaf counts and leaf instance masks. Table 1 summarizes the statistics of the CVPPP dataset, which exhibits the diversity of the number of images and average leaf counts in different domains.

Table 1: CVPPP 2017 dataset statistics under different domains. The number of images and average leaf counts are shown in the format of [training set / test set], while A5 exists only in the test set.

	A1	A2	A3	A4	A5
Resolution	500 × 530	530 × 565	2448 × 2048	441 × 441	Mixed
Num. of imgs	128 / 33	31 / 9	27 / 56	624 / 168	235
Avg. leaf counts	16.3 / 17.1	9.3 / 11.1	5.4 / 6	13.7 / 13.2	11.4

Leaf Disease Classification. The Cassava Leaf Disease Classification Dataset (denoted as Cassava below) [32] is a collection of real-world field images captured in Uganda, aimed at identifying various diseases found in Cassava leaves. The training set of the dataset comprises 21,397 images with a resolution of 800×600 pixels, where each image is categorized into one of the five classes representing the health status of the Cassava leaf: Cassava Bacterial Blight (CBB), Cassava Brown Streak Disease (CBSD), Cassava Green Mottle (CGM), Cassava Mosaic Disease (CMD), and Healthy (H). As shown in Table 2, the Cassava dataset exhibits significant class imbalance, where the images classified to CMD occupies over 60% of the dataset. The test set contains around 15,000 images held on Kaggle.

Table 2: The number of training images under different classes of the Cassava dataset.

	CBB	CBSD	CGM	CMD	H
Num. of imgs	1087	2189	2386	13158	2577

3.2. Experimental Setup

Figure 1 illustrates the model architecture and difference between the tuning methods, with details explained below.

Model Components. The models evaluated in this study are comprised two main components as shown in Figure 1 – a pre-trained backbone and a task-specific decoder. The backbone is a ViT-base feature extractor pre-trained using MAE, DINO, or DINOv2. The decoder for leaf counting and disease classification is a linear layer placed upon the backbone to regress the number of leaves or predict the classes of diseases. The decoder for leaf segmentation is adopted from ViTDet [33], consisting of a Feature Pyramid Network (FPN) [35] to generate multi-scale features and a Mask RCNN [22] decoder to predict instance masks.

Fine-tuning Methods. In our experiments, two distinct configurations for each model is implemented, namely decoder tuning and adapter tuning. Under both configurations, the pre-trained weights of the ViT-base backbone are kept frozen. During decoder tuning, only the parameters introduced by the decoder are trained. This setup examines the direct adaptability of the foundation models, which has not been exposed to the specific plant phenotyping datasets, with minimum tuning effort. On the other hand, in adapter tuning, LoRA is employed to fine-tune the backbone, so the parameters added by both LoRA and the decoder are trainable. This configuration investigates whether efficient fine-tuning of the pre-trained backbone can yield superior performance compared to decoder tuning. In addition, the evaluated models are compared with SoTA models that are specifically designed or trained for each task.

4. Experiments

Here, we discus experimental details and results on the three evaluated tasks. Decoder tuning is denoted as DT, and the evaluated models are reported in the form of "model-tuning method", where model $\in$ {MAE, DINO, DINOv2} and tuning method $\in$ {DT, LoRA}. All three foundation models are evaluated on leaf counting and disease classification, while only MAE and DINO are examined on leaf segmentation. Following the best practice [27], when using LoRA, only the query and value projection matrices in the attention blocks of a ViT are updated, with the rank set to 8.

4.1. Leaf Counting

Training setup. All the images from domain A1 to A4 of the CVPPP 2017 LCC training set are combined together to form a new training set named Ac. Ac is further divided into four equal sections for four-fold cross-validation (*i.e.* 75% training and 25% validation in each fold).

Mean Squared Error (MSE) loss function and Adam optimizer [30] are used to train the leaf counting models. Most models are trained with batch size of 16 and learning rate of 10^{-4}, except MAE-DT uses a learning rate of 5×10^{-3}. The input images are all resized to 448×448, with random rotation ($0° - 170°$) and flips applied for data augmentation.

Evaluation setup. The same metrics as the official LCC are used for evaluation, which are Difference in Count (DiC), Absolute Difference in Count (|DiC|), Mean Squared Error (MSE) and Percentage Agreement (PA). Assuming N is the total number of the evaluated images, $\hat{y}_i$ is the predicted leaf count on image i and y_i is the corresponding ground truth, these metrics are defined as:

$$\mathrm{DiC} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i),\ |\mathrm{DiC}| = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|,$$

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2, \quad \mathrm{PA} = \frac{100\%}{N}\sum_{i=1}^{N}\mathbf{1}[\hat{y}_i = y_i].$$

During inference, all the model predictions are rounded up to integers to match the nature of counting. Four-fold cross validation is performed on Ac and the average performance of the best model (obtaining the lowest validation MSE) of each fold is reported. During testing, the best models obtained in the validation are ensembled by averaging their predictions on the official test set.

Results. The validation set results from cross validation are shown in Table 3. From this table, we observe that using LoRA can help generalize foundation models to leaf counting because LoRA consistently outperforms DT. For example, the validation MSE is significantly improved from 1.4 to 0.95 for MAE, 1.25 to 0.88 for DINO, and 1.56 to 0.83 for DINOv2, respectively, after using LoRA. Over all the models, DINOv2-LoRA obtains the best performance.

Table 3: LCC four-fold cross validation results. DiC and |DiC| are reported in [mean (std)]. The best performance w.r.t. each metric is highlighted in **bold**.

	DiC ↓	|DiC| ↓	MSE ↓	PA [%] ↑
MAE-DT	-0.03 (1.18)	0.83 (0.84)	1.40	39.1
MAE-LoRA	0.01 (0.97)	0.67 (0.71)	0.95	44.9
DINO-DT	-0.11 (1.11)	0.81 (0.77)	1.25	37.7
DINO-LoRA	0.04 (0.94)	0.64 (0.69)	0.88	45.4
DINOv2-DT	0.01 (1.24)	0.93 (0.83)	1.56	32.0
DINOv2-LoRA	**0.01 (0.91)**	**0.62 (0.67)**	**0.83**	**47.6**

Table 4: LCC test results. The best performance under each domain is highlighted in **bold**. For DiC and |DiC|, results of A1 to A5 are reported in mean values only for better visualization, while the average performance across all domains (A1 + A2 + ... + A5) is reported under the column "All" in [mean (std)].

	A1	A2	A3	A4	A5	All
[14]	-0.39	-0.78	0.13	0.29	0.25	0.19 (1.24)
MAE-DT	-0.48	**0.11**	**0.0**	0.33	0.25	0.2 (1.89)
MAE-LoRA	-0.7	-0.56	-0.45	0.41	0.18	0.12 (1.33)
DINO-DT	-0.61	-1.0	-0.89	0.46	**0.12**	**0.05 (1.65)**
DINO-LoRA	0.24	-0.56)	-0.54	0.35	0.14	0.13 (1.37)
DINOv2-DT	-0.18	-0.78	-0.7	**0.02**	-0.16)	-0.17 (1.38)
DINOv2-LoRA	**-0.03**	-0.44	0.32	0.11	0.17	0.14 (1.27)

(a) DiC ↓

	A1	A2	A3	A4	A5	All
[14]	0.88	1.44	1.09	**0.84**	**0.90**	**0.91 (0.86)**
MAE-DT	1.03	1.67	2.18	1.08	1.34	1.33 (1.35)
MAE-LoRA	0.88	1.0	**1.05**	0.99	1.0	0.99 (0.89)
DINO-DT	1.21	1.44	1.64	1.11	1.23	1.24 (1.09)
DINO-LoRA	**0.79**	0.78	1.14	0.98	1.0	0.99 (0.95)
DINOv2-DT	0.91	0.78	1.27	0.94	1.01	1.01 (0.95)
DINOv2-LoRA	0.82	**0.67**	1.32	0.88	0.97	0.96 (0.84)

(b) |DiC| ↓

	A1	A2	A3	A4	A5	All
[14]	1.48	3	2.38	**1.28**	**1.53**	**1.56**
MAE-DT	1.76	5.67	8.43	2.16	3.66	3.6
MAE-LoRA	1.36	1.67	**2.02**	1.78	1.8	1.79
DINO-DT	2.12	3.67	4.61	2.17	2.72	2.73
DINO-LoRA	1.33	1.22	2.68	1.73	1.91	1.88
DINOv2-DT	1.64	1.22	3.05	1.61	1.94	1.92
DINOv2-LoRA	**1.24**	**0.89**	2.64	1.39	1.65	1.63

(c) MSE ↓

	A1	A2	A3	A4	A5	All
[14]	33.3	11.1	30.4	**34.5**	**33.2**	32.9
MAE-DT	27.3	22.2	12.5	31	26.8	26.5
MAE-LoRA	33.3	33.3	30.4	29.8	30.6	30.5
DINO-DT	18.2	22.2	21.4	28	27.2	26.1
DINO-LoRA	39.4	**44.4**	**32.1**	32.7	**33.2**	**33.5**
DINOv2-DT	**42.4**	**44.4**	26.8	31.5	31.1	31.7
DINOv2-LoRA	33.3	**44.4**	17.9	**34.5**	31.1	31.1

(d) PA [%] ↑

The test set results are shown in Table 4. This table compares the performance of the evaluated models with the champion of CVPPP 2017 LCC [14], which is a fully fine-tuned ResNet50 [23]. For convenience, the test performance of different models is mainly discussed based on MSE (Table 4 (c)). We first discuss the average performance over all test domains (*i.e.* A1 + A2 + ... + A5), which is shown in the rightmost column of Table 4 (c). As shown, MAE-DT and DINO-DT only reach MSE of 3.6 and 2.73, respectively, which are much worse than that of [14], 1.56. On the contrary, DINOv2-DT obtains a much better MSE of 1.92. As DINOv2 is pre-trained on a curated dataset that is larger than the one that DINO and MAE are pre-trained on, this indicates that increasing the scale of pre-trained source with carefully selected samples improves adaptability of foundation models. Similar to the observation in cross validation, using LoRA improves all the evaluated models, resulting in comparable MSE with [14] – 1.79, 1.88 and 1.63 for MAE, DINO and DINOv2, respectively. Overall, DINOv2-LoRA outperforms the other evaluated models, achieving similar performance with [14].

Note that LoRA dramatically improves the performance on A2 and A3 compared to DT, especially for MAE and DINO. In Table 4 (c), the 3rd column (A2) shows that the MSE drops from 5.67 to 1.67 for MAE and 3.67 to 1.22 for

DINO after applying LoRA; the 4th column (A3) presents that the MSE drops from 8.43 to 2.02 for MAE and 4.61 to 2.68 for DINO after applying LoRA. As shown in Table 1 (Section 3), A2 and A3 contain fewer images than the other domains. Also, A3 consists of tobacco images, which are different from the other domains (Arabidopsis) and have fewer leaf counts per plant. This observation indicates that LoRA is a potential solution to the scenarios of lacking data and domain shifts, both typical in plant phenotyping.

The findings in the LCC experiments are summarized as:

1. Fine-tuning foundation models using LoRA consistently outperforms DT, among which DINOv2-LoRA is the best, achieving similar performance with the SoTA task-specific model.

2. LoRA may be able to mitigate the effects of data scarcity and domain shifts.

4.2. Leaf Segmentation

Training setup. The training and validation splits in LSC are identical as LCC. Following the best practice [33], cross entropy and ℓ_1 loss are applied to mask predictions and bounding box regression, respectively. All the models are trained with the batch size of 16 and an Adam optimizer with learning rate of 10^{-4}. The input images are all resized to 448×448 with random flip applied as data augmentation.
Evaluation setup. The same metrics as the official LSC are used to evaluate different models, which are Best DICE score among all leaves (BestDice), DICE on foreground masks (FgBgDice), difference in leave count (DiffFG) and absolute difference in leaf count (|DiffFG|). Here, DiffFG and |DiffFG| function the same as DiC and |DiC| in LCC, but are computed based on the number of instance masks.

Similar to LCC, four-fold cross validation is performed on LSC and the average performance of the best model (achieving the highest validation BestDice) of each fold is reported. The predictions on the test set are ensembled by integrating the predicted instance masks on each leaf from all the best models obtained in the cross validation.
Results. Cross validation results are shown in Table 5: all the evaluated models obtain high BestDice (close or equal 0.9) and FgBgDice (over 0.9), indicating their capability of correctly segmenting leaves. On the other hand, the relatively worse DiffFG and |DiffFG| (over 2) shows that these models fail to detect all the presented leaves. For reference, the *worst* |DiffFG| obtained in the previous LCC experiments (Table 3) is 0.93. Comparing the performance between decoder tuning and LoRA on the same model, we observe that using LoRA does not increase the overall segmentation accuracy (BestDice and FgBgDice), but slightly improves the counting performance.

Figure 2 shows visual results on validation set samples from A1 to A4, which again verifies the findings in Table 5. In general, the segmentation quality of the evaluated models seem to be close to the ground truth, and it is difficult to tell the difference between DT and LoRA. The tendency to miss leaves and parts of them (*e.g.* the stem) are also evident.

Table 6 compares the test set results of the evaluated models against a SoTA model [20], which consists of

Table 5: LSC four-fold cross validation results. All metrics are reported in [mean (std)]. The best performance w.r.t. each metric is highlighted in **bold**.

	BestDice ↑	FgBgDice ↑	DiffFG ↓	|DiffFG| ↓
MAE-DT	**0.9 (0.05)**	**0.94 (0.05)**	-2.41 (2.18)	2.46 (2.12)
MAE-LoRA	**0.9 (0.05)**	**0.94 (0.05)**	**-2.35 (2.3)**	2.42 (2.23)
DINO-DT	0.88 (0.07)	0.91 (0.08)	-2.78 (2.19)	2.81 (2.15)
DINO-LoRA	0.88 (0.07)	0.92 (0.07)	-2.36 (1.98)	**2.39 (1.94)**

Table 6: LSC test results. The best performance under each domain is highlighted in **bold**. Results of A1 to A5 are reported in mean values only for better visualization, while the average performance across all domains (A1 + A2 + ... + A5) is reported under the column "All" in [mean (std)].

	A1	A2	A3	A4	A5	All
[20]	**0.92**	**0.9**	**0.92**	**0.89**	**0.9**	**0.9**
MAE-DT	0.90	0.87	0.83	**0.89**	0.88	0.88 (0.08)
MAE-LoRA	0.89	0.87	0.81	**0.89**	0.87	0.87 (0.1)
DINO-DT	0.85	0.80	0.71	0.85	0.82	0.82 (0.13)
DINO-LoRA	0.85	0.82	0.74	0.85	0.82	0.82 (0.12)

(a) BestDice ↑

	A1	A2	A3	A4	A5	All
[20]	0.94	0.88	0.91	**0.94**	0.93	0.93
MAE-DT	**0.96**	**0.92**	0.94	**0.94**	0.94	0.94 (0.04)
MAE-LoRA	**0.96**	**0.92**	**0.95**	**0.94**	**0.95**	**0.95 (0.04)**
DINO-DT	0.94	0.89	0.88	0.93	0.92	0.92 (0.1)
DINO-LoRA	0.95	0.90	0.90	0.93	0.93	0.93 (0.08)

(b) FgBgDice ↑

	A1	A2	A3	A4	A5	All
[20]	-2	-2.11	-1.7	**-0.92**	-1.16	-1.21
MAE-DT	-1.30	**-1.00**	-0.71	-1.76	-1.49	-1.47 (1.81)
MAE-LoRA	-1.27	-1.11	-0.64	-1.56	-1.33	-1.32 (1.82)
DINO-DT	-1.70	-1.22	**-0.02**	-1.74	-1.32	-1.34 (2.04)
DINO-LoRA	**-1.00**	-1.22	-0.25	-1.23	**-0.99**	**-0.99 (1.88)**

(c) DiffFG ↓

	A1	A2	A3	A4	A5	All
[20]	2.06	2.11	1.73	**1.12**	**1.31**	**1.36**
MAE-DT	**1.36**	1.44	**1.39**	1.80	1.68	1.66 (1.63)
MAE-LoRA	1.45	1.33	1.68	1.63	1.63	1.62 (1.56)
DINO-DT	1.82	1.44	1.88	1.81	1.82	1.81 (1.63)
DINO-LoRA	1.55	**1.22**	1.79	1.54	1.58	1.58 (1.41)

(d) |DiffFG| ↓

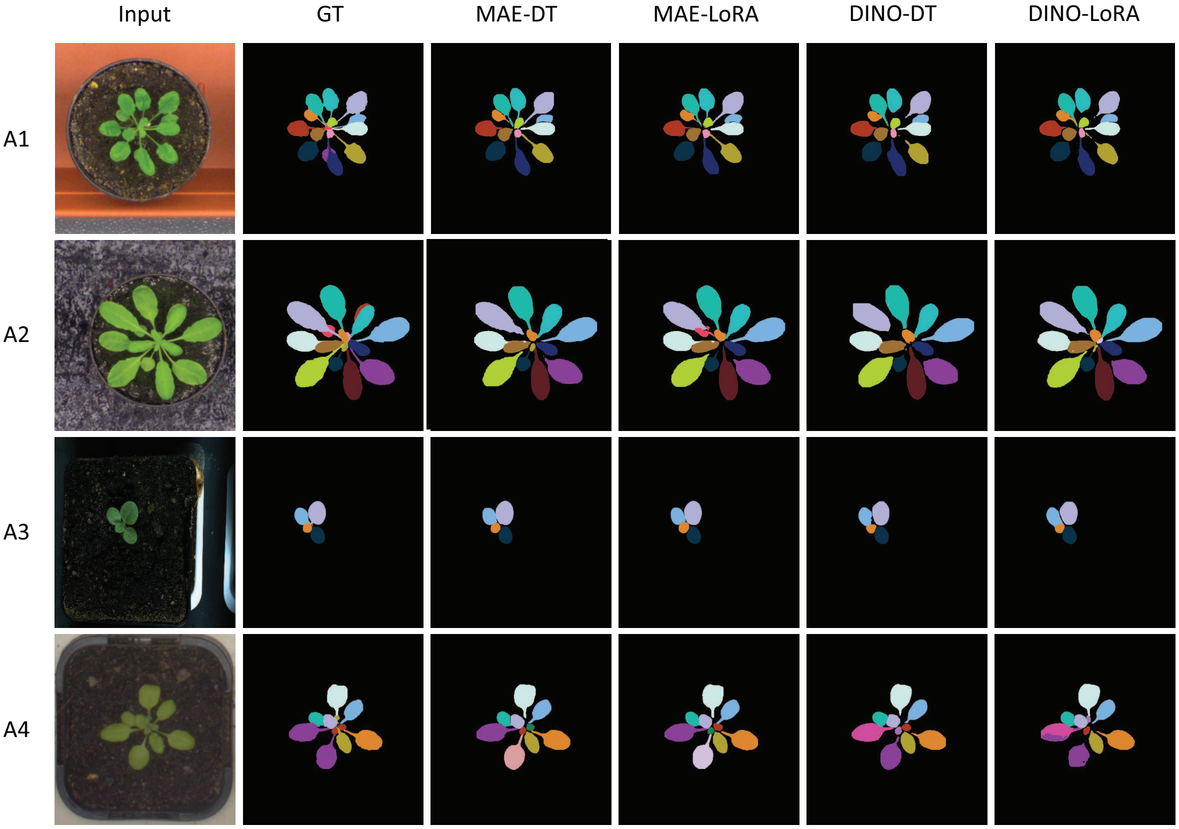


Figure 2: Visual results on LSC validation set. Samples from A1 to A4 are shown in different rows. The original images, ground-truth masks, and model predictions of the samples are shown in the first, second, and the remaining columns, respectively. The evaluated models miss some (small) leaves or stems, as observed in most samples.

ResNet101, FPN and DeepLab V3+ [8] . Comparing the BestDice over all test domains (Table 6 (a) rightmost column), we find that MAE-DT and MAE-LoRA perform closely to [20], with only 0.02 and 0.03 difference, while DINO-DT and DINO-LoRA underperform. This sub-table also shows that LoRA does not improve the evaluated models in leaf instance segmentation compared to DT.

Note that both MAE and DINO underperform in A3 (Table 6 (a) fourth column), with about 0.05 and 0.1 BestDice drop compared to their average performance. This may be caused by downsampling the inputs. This resizing effect is also a potential cause of missing leaves as observed in validation. Besides, in both validation and testing, we do not observe improvement after using LoRA. This may be caused by the large difference between the number of trainable parameters in the FCN + Mask RCNN decoder (20% of the whole model) and those added by LoRA (0.34% of the whole model).

The findings of the LSC experiments are summarized as:

1. Adapting MAE for leaf instance segmentation achieves slightly worse results than [20], while the fine-tuned DINO models perform much worse.

2. LoRA does not outperform DT potentially due to the heavy imbalance between the number of parameters added by the decoder and LoRA.

3. The evaluated models can miss leaves or stems, although this may be caused by resizing of the inputs.

4.3. Leaf Disease Classification

Training setup. Following the first place solution [18] in the Cassava Leaf Disease Classification competition, the Bi-Tempered Logistic Loss [1] and Adam optimizer are used to train the classification model. MAE-DT is trained with learning rates starting from 10^{-2} and then down-scaling by a factor of 10 at epoch 25 and 50, while all the other models are trained with a consistent learning rate of 10^{-4}. The training images are first randomly cropped and then resized to 448×448, followed by a series of random transformation as used in [18].

Evaluation setup. Per-class accuracy is reported in our validation set, while only overall accuracy is reported in the official test set. Five-fold cross validation is performed and the average performance of the best models (achieving the best overall accuracy) obtained in each fold is reported. During testing, the best models obtained in the cross validation are ensembled by averaging their predicted scores.

Results. Table 7 presents the cross validation results on the validation set. The table shows that LoRA consistently improves model generalization on the Cassava leaf disease classification task compared to DT. After applying LoRA, the overall accuracy (the rightmost column) is raised from 77.9% to 88.2% on MAE, 85% to 88.2% on DINO, and 86.6% to 90.3% on DINOv2. In general, DINOv2-LoRA outperforms all the other evaluated models on every class, while MAE-DT performs the worst.

Table 7: Cross validation accuracy [%] on Cassava disease classification. The column "All" reports the average accuracy across all classes. The highest accuracy under each class is highlighted in **bold**.

	CBB	CBSD	CGM	CMD	H	ALL
MAE-DT	37.1	50.4	47.3	94.7	61.1	77.9
MAE-LoRA	59.6	80.3	76.1	97.2	72.4	88.2
DINO-DT	51.9	73.3	70.0	95.6	68.9	85.0
DINO-LoRA	60.9	79.1	78.1	97.0	72.1	88.2
DINOv2-DT	57.9	73.7	70.4	96.7	74.1	86.6
DINOv2-LoRA	**67.0**	**82.1**	**81.9**	**97.6**	**76.9**	**90.3**

As Table 2 (Section 3) shows, the number of training images in the CMD class is significantly larger than that of the remaining classes. This severe class imbalance is directly reflected in the cross validation results as shown in Table 7, where all the evaluated models exhibit supreme accuracy on CMD. However, LoRA shows potential in addressing class imbalance. Comparing MAE-DT and MAE-LoRA, the accuracy is dramatically improved from 36.3% to 59.6%, 50.7% to 80.3%, 46.3% to 76.1%, and 62.4% to 72.4% on CBB, CBSD, CGM and H, respectively. The performance of DINO-LoRA and DINOv2-LoRA also shows clear improvement on these minority classes.

Table 8 compares the test set accuracy of the champion of the Cassava Plant Disease Classification competition [18], which ensembles ResNext50 [48], ViT-B, EfficientNet B4 [45] and MobileNet V3 [26], with that of the evaluated models. The observation in the test set is similar to that of the validation set: LoRA consistently improves the model adaptation, with DINOv2-LoRA performing the best, but slightly worse than [18]. Note that [18] ensembles four different architectures while we only use ViT-B.

Table 8: Test set accuracy [%] on the Cassava dataset.

Method	Accuracy
[18]	**91.3**
MAE-DT	77.2
MAE-LoRA	88.8
DINO-DT	83.9
DINO-LoRA	89
DINOv2-DT	86.1
DINOv2-LoRA	89.7

In summary, the findings in the Cassava experiments are:

1. LoRA consistently enhances model adaptation in leaf disease classification. Among the evaluated models, DINOv2-LoRA performs the best, achieving slightly worse performance than SoTA.

2. LoRA shows potential of addressing class imbalance.

5. Conclusion

This paper demonstrates the possibility of adapting a foundation model for multiple plant phenotyping tasks through extensive experiments. Among the evaluated foundation models, DINOv2-LoRA achieves the best performance in leaf counting and disease classification, which is close to that of the SoTA model for each task. In leaf segmentation, MAE outperforms DINO, approaching the SoTA level. In general, we find that each foundation model can be efficiently adapted for all the three tasks with acceptable performance.

In terms of the tuning methods, LoRA consistently outperforms decoder tuning in leaf counting and disease classification. Moreover, LoRA exhibits the potential of mitigating issues related to data scarcity, domain shifts, and class imbalance. However, in leaf segmentation, LoRA fails to further improve the foundation models over decoder tuning. This may be due to the segmentation decoder introducing notably more parameters than LoRA.

In summary, our work pioneers systematic assessment of the adaptation of foundation models for plant phenotyping tasks. It not only serves as an essential benchmark in this field but also discusses the optimal use-cases of different foundation models and tuning methods. The continued development of general-purpose models and adapters should draw more attentions of the plant phenotyping community.

Acknowledgements

This project was funded by the BBSRC grant BB/Y512333/1 "PhenomUK-RI: The UK Plant and Crop Phenotyping Infrastructure".

References

[1] Ehsan Amid, Manfred KK Warmuth, Rohan Anil, and Tomer Koren. Robust bi-tempered logistic loss based on bregman divergences. *Advances in Neural Information Processing Systems*, 32, 2019. 7

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 3

[3] Jonathan Bell and Hannah M. Dee. Aberystwyth leaf evaluation dataset [data set], 2016. 3

[4] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 1, 2

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 2, 3

[7] Feng Chen, Michael P Pound, and Andrew P French. Learning to localise and count with incomplete dot-annotations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 1612–1620, 2021. 1

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 7

[9] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022. 2, 3

[10] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022. 2, 3

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 3

[13] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023. 2

[14] Andrei Dobrescu, Mario Valerio Giuffrida, and Sotirios A Tsaftaris. Leveraging multiple datasets for deep leaf counting. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 2072–2079, 2017. 1, 5

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[16] Raman Dutt, Linus Ericsson, Pedro Sanchez, Sotirios A Tsaftaris, and Timothy Hospedales. Parameter-efficient finetuning for medical image analysis: The missed opportunity. *arXiv preprint arXiv:2305.08252*, 2023. 2, 3

[17] Mario Valerio Giuffrida, Massimo Minervini, and Sotirios A Tsaftaris. Learning to count leaves in rosette plants. In *Proceedings of the British Machine Vision Conference Workshops*, 2015. 1

[18] Team golddiggaz. First place solution to the cassava leaf disease classification [competition solution]. https://www.kaggle.com/competitions/cassava-leaf-disease-classification/discussion/221957. 7, 8

[19] Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19338–19347, 2023. 2

[20] Ruohao Guo, Liao Qu, Dantong Niu, Zhenbo Li, and Jun Yue. Leafmask: Towards greater accuracy on leaf segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1249–1258, 2021. 1, 6, 7

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 2, 3

[22] 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. 4

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[24] Sheng He, Rina Bao, Jingpeng Li, P Ellen Grant, and Yangming Ou. Accuracy of segment-anything model (sam) in medical image segmentation tasks. *arXiv preprint arXiv:2304.09324*, 2023. 2

[25] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 2, 3

[26] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 8

[27] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2, 3, 4

[28] Ge-Peng Ji, Deng-Ping Fan, Peng Xu, Ming-Ming Cheng, Bowen Zhou, and Luc Van Gool. Sam struggles in concealed scenes–empirical study on" segment anything". *arXiv preprint arXiv:2304.06022*, 2023. 2

[29] Wei Ji, Jingjing Li, Qi Bi, Wenbo Li, and Li Cheng. Segment anything is not always perfect: An investigation of sam on different real-world applications. *arXiv preprint arXiv:2304.05750*, 2023. 2

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[31] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2, 3

[32] Makerere Artificial Intelligence (AI) Lab. Cassava leaf disease classification [data set]. `https://www.kaggle.com/competitions/cassava-leaf-disease-classification`. 4

[33] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pages 280–296. Springer, 2022. 4, 6

[34] Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*, 2023. 2

[35] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4

[36] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE transactions on knowledge and data engineering*, 35(1):857–876, 2021. 2

[37] Massimo Minervini, Andreas Fischbach, Hanno Scharr, and Sotirios A Tsaftaris. Finely-grained annotated datasets for image-based plant phenotyping. *Pattern recognition letters*, 81:80–89, 2016. 3

[38] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 3

[39] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022. 3

[40] Michael P Pound, Jonathan A Atkinson, Darren M Wells, Tony P Pridmore, and Andrew P French. Deep learning for multi-task plant phenotyping. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2055–2063, 2017. 1

[41] Chao Qi, Murilo Sandroni, Jesper Cairo Westergaard, Ea Høegh Riis Sundmark, Merethe Bagge, Erik Alexandersson, and Junfeng Gao. In-field classification of the asymptomatic biotrophic phase of potato late blight based on deep learning and proximal hyperspectral imaging. *Computers and Electronics in Agriculture*, 205:107585, 2023. 1

[42] Muhammad Hammad Saleem, Johan Potgieter, and Khalid Mahmood Arif. Plant disease detection and classification by deep learning. *Plants*, 8(11):468, 2019. 1

[43] Hanno Scharr, Massimo Minervini, Andreas Fischbach, and Sotirios A Tsaftaris. Annotated image datasets of rosette plants. In *European conference on computer vision. Zürich, Suisse*, pages 6–12, 2014. 3

[44] Abraham George Smith, Jens Petersen, Raghavendra Selvan, and Camilla Ruø Rasmussen. Segmentation of roots in soil with u-net. *Plant Methods*, 16(1):1–15, 2020. 1

[45] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 8

[46] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022. 3

[47] Junde Wu, Rao Fu, Huihui Fang, Yuanpei Liu, Zhaowei Wang, Yanwu Xu, Yueming Jin, and Tal Arbel. Medical sam adapter: Adapting segment anything model for medical image segmentation. *arXiv preprint arXiv:2304.12620*, 2023. 3

[48] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 8

[49] Robail Yasrab, Jonathan A Atkinson, Darren M Wells, Andrew P French, Tony P Pridmore, and Michael P Pound. Rootnav 2.0: Deep learning for automatic navigation of complex plant root architectures. *GigaScience*, 8(11):giz123, 2019. 1

[50] Yuxuan Yuan, Bao Linh Ton, William J. W. Thomas, Jacqueline Batley, and David Edwards. Supporting crop plant resilience during climate change. *Crop Science*, 63(4):1816–1828, 2023. 1