Skoltech

Skolkovo Institute of Science and Technology

MASTER'S THESIS

Plant phenotyping using deep learning methods in a data-poor environment

Master's Educational Program: Data Science

Student

Liliya Lemikhova
Data Science
June 10, 2022

Research Advisor:

Andrey Somov
Associate Professor

Moscow 2022

Skoltech

Skolkovo Institute of Science and Technology

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Фенотипирование растений с использованием методов глубокого обучения в условиях малого количества данных

Магистерская образовательная программа: Науки о данных

Студент_____

Лилия Лемихова

Науки о данных

Июнь 10, 2022

Научный руководитель:_____

Андрей Сомов

Профессор

Москва 2022

Plant phenotyping using deep learning methods in a data-poor environment

Liliya Lemikhova

Submitted to the Skolkovo Institute of Science and Technology
on June 10, 2022

Abstract

Plant phenotyping is one of the most demanded tasks in digital agriculture. A common approach to solving this problem is image analysis using computer vision methods Thanks to the rapid development of computing technologies and deep learning methods, neural networks today solve many computer vision problems with an accuracy that surpasses that of a human. However, for accurate results, they require lots of data.

Plant datasets are hard to collect and label; hence, datasets are usually small. In this work, we examine different strategies for dealing with the lack of data for two types of plant phenotyping task - classification and object detection.

In the classification part, we apply computer vision for plant recognition at the Antarctic station greenhouse, a training facility for future space colonization missions. Our experiments rely on transfer learning and explore the importance of the pre-training data domain. We show that a common approach of using models pre-trained on the Imagenet dataset can be further improved using publicly available domain-specific datasets. The classification results of 16 plant varieties with the ResNet50 model increase the F-score from 75% to 82% using only three training images. We also achieve 78% top-3 accuracy without any training data.

In the object detection part, a recently proposed method for virtual outliers synthesis is applied for out-of-distribution fruits detection. We demonstrate that this method allows to increase AUROC from 70.8% to 77.2% without any damage to mAP and without any additional training data. We analyze for which datasets this method is suitable, and for which it requires modification. We demonstrate the drawbacks of the evaluation method for specific kinds of datasets and suggest improvements.

In conclusion, we suggest best practices for how to deal with small agricultural datasets in various conditions for zero- and few-shot classification and for out-of-distribution detections.

Research Advisor:
Name: Andrey Somov
Degree: PhD
Title: Associate Professor

Фенотипирование растений с использованием методов глубокого обучения в условиях малого количества данных

Лилия Лемихова

Представлено в Сколковский институт науки и технологий
Июнь 10, 2022

Реферат

Фенотипирование растений - одна из наиболее распространенных задач в области цифрового сельского хозяйства. Один и способов решения этой задачи - анализ изображений растений с помощью методов компьютерного зрения. Благодаря быстрому развитию вычислительных технологий и методов глубокого обучения, нейронные сети сегодня позволяют решать задачи компьютерного зрения с точностью превосходящий человеческую. Однако для точных результатов им требуются датасеты с большим количеством изображений. Датасеты с изображениями растений трудно собирать и размечать; как результат, датасеты обычно содержат мало изображений. В этой работе рассматриваются различные стратегии решения проблемы нехватки данных для двух задач компьютерного зрения для фенотипирования растений - классификации и детекции объектов.

Задача классификации для малых датасетов рассматривается на примере задачи распознавания растений в теплице Антарктической станции, учебном центре для будущих миссий по колонизации космоса. Наши эксперименты основаны на методах трансферного обучения и исследуют важность выбора датасета для предобучения. Мы показываем, что привычный подход с использованием моделей, предварительно обученных на датасете Imagenet, может быть дополнительно улучшен с использованием открытых датасетов, специфичных для конкретной предметной области. Результаты классификации 16 сортов растений с помощью модели ResNet50 увеличивают F-меру с 75% до 82%, используя только по 3 изображения на каждый класс в обучающей выборке. Мы также достигаем 78% точности на топ-3 без использования данных из целевого датасета в обучении.

Проблема нехватки данных для задачи обнаружения объектов рассматривается на примере задачи детекции объектов нецелевых классов для датасета с фруктами. Для этого мы применяем недавно предложенный метод синтеза виртуальных выбросов. Мы показываем, что этот метод позволяет увеличить AUROC с 70,8% до 77,2% без какого-либо ущерба для mAP и без каких-либо дополнительных данных. Мы анализируем для каких датасетов данный метод подойдет, а для каких требует модификации. Кроме того, мы демонстрируем недостатки метода оценивания и предлагаем способы его усовершенствовать.

В заключение, мы предлагаем стратегии для лучшей работы в условиях малого количества данных в зависимости от ситуации для рассмотренных задач компьютерного зрения в области сельского хозяйства.

Научный руководитель:
Имя: Андрей Сомов
Ученое звание, степень: Доктор физ.-мат. наук
Должность: Профессор

Contents

1 Introduction **8**

2 Related works **11**

2.1 Classification 11

2.2 OOD detection 12

3 Few-shot plant phenotyping:

Antarctic station use case **13**

3.1 Methods 13

3.1.1 Zero-shot Transfer 13

3.1.2 k-shot Learning 14

3.2 Experiments and Results 15

3.2.1 Metrics 15

3.2.2 Datasets 15

3.2.3 Zero-shot Classification 15

3.2.4 Few-shot Classification 17

4 Out-of-distribution (OOD) detection with fruit dataset **22**

4.1 Methods 22

4.2 Experiments and results 24

4.2.1 Data 24

4.2.2 Evaluation 27

4.2.3 Experiment details 27

4.2.4 Results and discussion 27

5 Conclusions **35**

5.1 Classification 35

5.1.1 Zero-shot 35

5.1.2 Few-shot 35

5.2 Object detection 36

List of Figures

3.1 EDEN ISS Mobile Test Facility inside. 16

3.2 Metrics for classification in zero-shot scenario. 18

3.3 F1-scores of classification for k-shot scenario depending on number of training examples per class. 20

3.4 Confusion matrix of predictions by 5-shot CLIP ViT/16. The model confuses loose-leaf lettuce (label ”4”) and mizuna lettuce (label ”5”). 21

4.1 Umap visualisation of all classes. 25

4.2 ROC and Precision-Recall curves for Deep Fruits, Open Images and COCO datasets by models with and without VOS method. 28

4.3 Dataset of ID classes. Left column: results by baseline model; right column: results by the VOS method. 32

4.4 Dataset of OOD classes. Left column: results by baseline model; right column: results by the VOS method. 33

4.5 Examples of detections that become problematic for calculating FPR on COCO dataset. The pictures on the left are from ID dataset; thus all detections, including false positive ones, are considered to be ID. The pictures on the right are from OOD dataset; thus all detections, including true positive ones, are considered to be OOD. 34

List of Tables

3.1 Results for zero-shot classification 18
3.2 F1-scores of k-shot classification with $k \in \{1, 3, 5\}$. 20

4.1 Number of bounding boxes and its type for each fruits class fruits datasets. 26
4.2 Number of images in used datasets. 27
4.3 Comparative results of the models trained with and without VOS method for test fruits datasets made from COCO, Open Images and Deep Fruits images. 29

Chapter 1

Introduction

The development of technological capabilities and methods of deep learning has greatly influenced many industries, including agriculture [46][3]. Computer vision application is one of the main directions of development [47]. The accuracy of modern image analysis methods often exceeds the accuracy of a human. A more advanced approach to the care for crops can potentially make it possible to use resources more efficiently, harvest faster and more accurately, prevent the development of diseases in the early stages, and, as a result, use fewer herbicides and pesticides harmful to environment and humans [59]. In addition, agricultural conditions such as open-air huge fields and tall trees are often difficult for manual human labor. Thus, many studies are aimed at introducing computer vision methods into the field of agriculture [42][35][33].

Plant phenotyping is one of the most fundamental and demanded tasks in this area. It can be considered in the form of classification problem, if we need just to classify an object on the picture, or in the form of object detection problem, if in addition to classification we want to localize the objects by predicting a bounding box for each instance of target classes. Deep learning methods have shown their great potential and usability for both these problems; however, they usually require large labeled datasets [48].

Many studies use datasets in a controlled environment and with a simple background such as *PlantVillage* dataset [20]. Such datasets are rather easy to collect; however, such datasets differ greatly from the natural environment [39], and due to differences in light, noise, and background, the results are much worse when models are tested in real conditions.

Images of plants in real conditions are often hard to take. Sometimes the required plants are available only for a short period of time, for example blooming period lasts for a couple of weeks once a year. Also, the number of locations for collecting datasets is very limited and they are often situated in hard-to-reach places. Image annotation is another significant problem for plant phenotyping [38]. It is expensive and time-consuming. Moreover, it often requires professional knowledge to get accurate annotation. To overcome this challenge, we need to develop and test methods of few-shot learning.

Few-shot learning is a scenario when there are few labeled samples per class in a training

dataset [54]. Usually, few-shot classification means classification when each class has the same number of images - k. If there is no labeled data for the target dataset, such a case is called a zero-shot learning problem. There are three groups of few-shot methods, depending on which aspect is enhanced by prior knowledge [54]:

- *model* to constrain the hypothesis space;
- *data augmentation* to diversify the training set;
- *algorithm* to search for best parameterization by providing a good initialization.

In this study, we will focus on the last two groups of methods.

Deep learning researchers often ignore prior domain knowledge, which could result in a better initialization and, therefore, easier optimization task. Thus, reducing the sizes of required datasets. In this work, we perform a plant classification task at the Antarctic station greenhouse using computer vision methods. We use the transfer learning approach and investigate the essence of the pretraining data domain. In particular, we demonstrate that in some cases it is more efficient to use publicly available domain-specific datasets instead of commonly used ImageNet dataset.

The problem for object detection is that models perform generally quite successfully on test images from the same distribution as the training data; however, they often produce high scores for out-of-distribution (OOD) objects which should be ignored [40]. It becomes especially challenging to distinguish false positives when in-distribution (ID) and OOD objects are similar to each other as often happens in the precision agriculture domain. To overcome this issue, researchers typically add many images with objects which should not be detected during training. These objects are usually referred to as ”negative examples”. Although it usually helps, sometimes there is not enough or even no available data to use as negative examples during training. We demonstrate this problem on a fruit detection task. Suppose, there is a farm where different fruits are growing close to each other. To look after these plants, we use robotics computer vision systems. Different plants require different care such as irrigation and fertilization; so, we want to be sure that if we detected an object and assigned a class to it we did it correctly. If we start to grow a new type of plant, we want our model not to detect this new type as one of the ID classes. We do not want to collect new data and retrain our neural network. To achieve this, we use a recently proposed method that allows us to detect OOD objects and does not require additional data with negative examples.

The text is structured the following way: we start with a literature review and continue with experiments for solving plant phenotyping tasks - one for classification and another one for object detection. Each experimental part is presented similarly: first, we explain used methods, then,

describe experiment setup, and finally present results and discussions of the results. After that, we provide conclusions for both parts and perspectives on the future work.

Chapter 2

Related works

2.1 Classification

For image classification task, transfer learning is an appropriate technique in a data-poor environment [58]. In this scenario, we pretrain a model on a big dataset and then fine-tune it on a small target dataset. The most popular choice for a pretraining dataset is ImageNet dataset [25]. However, for fine-grained agricultural datasets, it might be not suitable enough and it has several drawbacks [50]. Most images are objects centric, the objects are large and differ strongly from each other in contrast to typical images from the agricultural domain [6].

In [52] the authors analyzed the performance of feature extractors trained with supervised and self-supervised methods on ImageNet and iNaturalist. They demonstrated that standard supervised methods still outperform self-supervised approaches. The model pretrained on the iNaturalist dataset performed worse than the model pretrained on ImageNet on all tested datasets; but, it showed better performance on the datasets related to nature: Flowers [41], CUB [55] and NABirds [51]. However, the authors did not study the effect of pretraining on iNaturalist in a few-shot scenario.

Another option for pretraining is to use a dataset consisting of (image, text) pairs, predicting which capture goes with each image. A recent successful implementation of this idea is a contrastive language-image pretraining model introduced in [43]. This approach performs particularly well for zero-shot prediction when we do not need any data for training.

In [1], the authors performed a transfer from a source large dataset to a small one. However, the source and target datasets were subsets from the same Plant village dataset with pictures of different leave diseases in the same conditions and simple background. There is usually no additional big dataset with images of the same domain in reality. In [30], the authors ran experiments for few-shot pests recognition with convolutional neural network feature extractor and triplet loss. This is a bit different domain as pests are usually easier to classify than plants. Moreover, the authors did not study the effect of the pretraining dataset.

2.2 OOD detection

We can define OOD detection as a problem of binary classification into in- and out-of-distribution objects. Out-of-distribution objects are the objects with semantics outside the support of the target semantic labels [10].

OOD detection study has been mostly developed for image classification problem. For image classification problem, there are mainly two approaches - OOD detection based on a special score calculated after the training stage and OOD detection based on regularization during the training stage. In [2], the authors use OpenMax score based on extreme value theory. Other methods for defining a score were developed such as to use maximum softmax probability [17] and use Mahalanobis distance [28]. The regularization-based methods use real outlier images [18] [37] or synthesize outliers with GANs in the pixel space [27].

However, this topic is undercovered for object detection task. The main method was to form an additional diverse dataset with OOD examples that could appear during the test stage. After that, run experiments using these images without bounding boxes annotation. However, such datasets are not always available or just are not varied enough to cover all possible unknown objects. In [23], the authors use energy-score [34] to find OOD objects; however, they do not optimize a model to detect OOD objects during training stage.

In the work [10], the authors suggested a new method how to recognise OOD objects. OOD samples are produced right in the feature space during training and the model is optimized simultaneously to detect ID samples and to distinguish that the detected object is OOD. This method does not require any additional data. However, the authors did not conduct any study on how this method works when ID and OOD classes are close to each other and it usually happens in the agricultural domain.

Chapter 3

Few-shot plant phenotyping: Antarctic station use case

3.1 Methods

Our goal is to study several scenarios in case of a lack of data for training. In this setting, there are two categories of methods - zero-shot and few-shot learning. For few-shot learning, we study a different number of samples per class. Thus, we will name this setting k-shot learning, where k is a non-zero number of training samples from each class.

3.1.1 Zero-shot Transfer

There are two approaches that we tried for this scenario:

- **Third-party data sources.** This is a common approach when there is no data for training to find images of the same classes and if possible from a similar domain. If a new dataset is too small, we can freeze the parameters of the model pretrained on another large dataset and train only the last layers. The first disadvantage of this approach is that there are not always publicly available required images. Moreover, it needs additional time and resources to pretrain the model on these new data.

- **Use of class names.** Contrastive Language-Image PreTraining (CLIP) is a recently published model that was trained on a dataset consisting of 400 million various (image, text) pairs collected from the internet [43]. It was trained to predict the caption for an image. If provided a set of possible classes, it can predict the most suitable one for an input image without any additional training.

3.1.2 k-shot Learning

To tune the model for the target dataset, depending on k, the dataset for pretraining, and the target dataset, we can use one of the following scenarios:

- freeze all parameters, add one or several linear layers with activation functions and learn only the weights of these final linear layers
- freeze part of deeper layers where layers identify general features of an image and fine-tune weights of last layers, which identify more task-specific features of an image.

For the target dataset, we conducted experiments with a maximum $k = 5$, so we used the first strategy for models pretrained on a large dataset.

In addition to this method, we further fine-tuned models pretrained for zero-shot learning. For this approach, we used warmup not to degrade learned weights with too big optimization steps [13]. We also use image augmentation as it has been shown to make models more accurate and stable, especially for small datasets [21, 22]. Finally, we tested how weights of pretrained CLIP perform as initial ones for a few-shot scenario.

3.2 Experiments and Results

3.2.1 Metrics

To evaluate results, we used the weighted F1-score metric [19]. To define F1-score, we first need to define two additional metrics: precision and recall:

$$precision = \frac{TP}{TP + FP}, \tag{3.1}$$

$$recall = \frac{TP}{TP + FN}, \tag{3.2}$$

where TP and TN are the numbers of true positive and true negative classifications, while FP and FN are the numbers of false-positive and false-negative classifications. Then F1 score is defined in the following way:

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall}. \tag{3.3}$$

The F1-score is calculated for each class and then averaged with weights inversely proportional to the number of ground truth images per class. For zero-shot classification we also calculated top-N accuracy, that means we assume prediction to be correct if the correct class is in the top-N probabilities.

3.2.2 Datasets

- **ImageNet1K** is a large dataset for classification with more than 1.2 million hand-annotated images with 1,000 synsets [7].
- **iNaturalist** dataset is a large publicly available fine-grained dataset with more than 800,000 images of more than 5000 species of plants and animals pictures taken in the wild [53].
- **Target dataset** consists of 228 images of size 1000×1000 with 16 plant classes and one extra class for empty slots. It was collected by webcams with top-down and side views that monitor the entire greenhouse (see Fig. 3.1).

3.2.3 Zero-shot Classification

In this scenario, we used two methods: getting data from an external open dataset for each class of the target dataset and using the CLIP model in zero-shot mode.

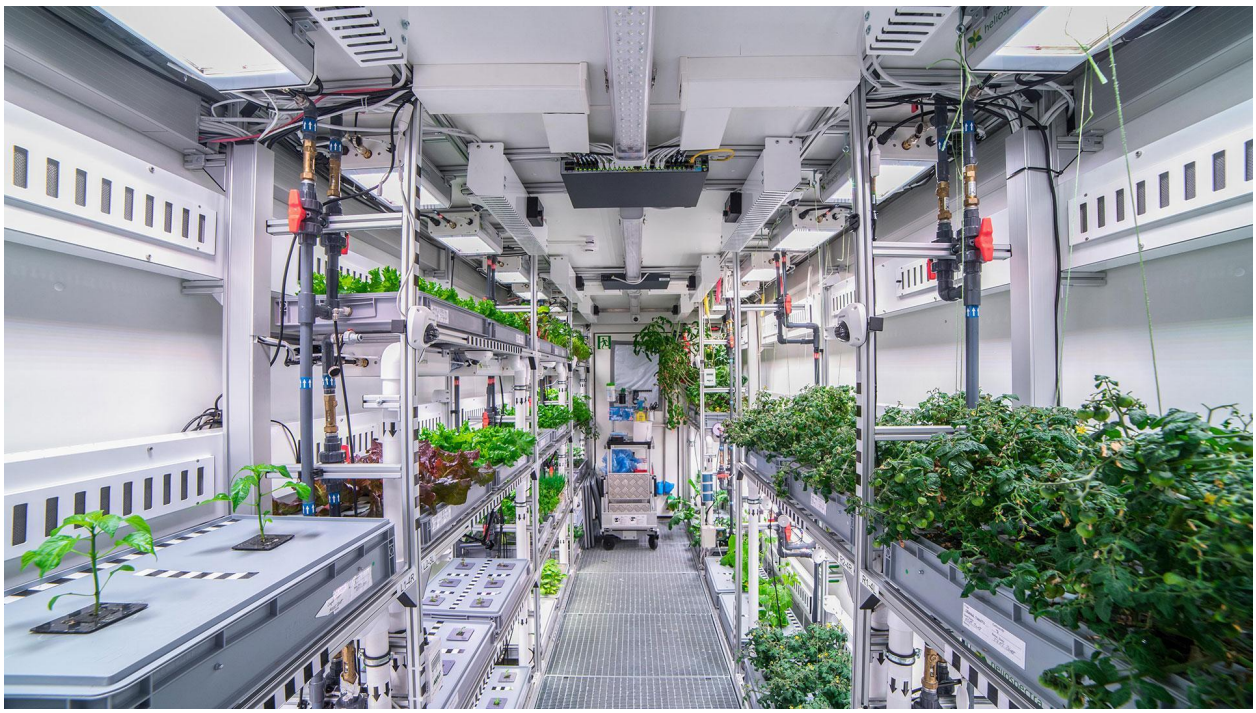

Figure 3.1: EDEN ISS Mobile Test Facility inside.

Getting more data.

For this approach, we needed to find more images close to classes from the target dataset. We tried two options:

- Images from iNaturalist dataset. We took 50 images of the same plant family from the iNaturalist dataset for each class of the target dataset except for the class of empty slots. For some classes of the target dataset the plant family from the iNaturalist dataset was the same; thus, to avoid repetition, we assigned images of this family to one of the classes from the target dataset and left others empty.

- Images retrieved by CLIP. For each class in the target dataset, we formed its textual description in the following form: "{name of class} plants in greenhouse". Then we got 100 images per class whose embeddings are the closest neighbors to the embedding of the corresponding textual description.

After collecting the dataset for pretraining, we need to train the model. First, we froze all layers except the last two convolutional layers of ResNet50 [16] pretrained on ImageNet. Then, we changed the output dimension of the last fully connected layer to be equal to the number of classes in the target dataset (17). Finally, fine-tuned parameters to classify the collected dataset using cross-entropy loss. We used batch-size of 64, $Adam$ optimizer [24] starting with learning rate $5 \cdot 10^{-4}$, $ReduceLROnPlateau$ scheduler and trained for 20 epochs. We used $RandomResizedCrop(224)$,

$RandomRotation$, $ColorJitter$ and $CutOut$ [8] for data augmentation for this approach and for experiments in few-shot classification section.

CLIP.

For each class, we formed its description in the following form: 'A photo of {name of class} plants in greenhouse'. After that, the model classifies an image depending on the closest class description in the embedding space.

Results and discussion.

The results of these approaches on target dataset are presented in Table 3.1 and in Fig. 3.2.

CLIP with Vision-Transformer [9] as a backbone performs best with all metrics without additional data, which is very convenient. However, the F1 score and top-1 accuracy values are not satisfactory enough to be competitive with few-shot learning results, opposite to what was stated in the paper. This decline in accuracy might be because the pretraining dataset for the CLIP model did not include enough samples from the domain of the target dataset. Thus, future work might be to train CLIP on a dataset that contains samples from the agricultural domain to improve results for zero-shot learning. Additionally, the choice of form for textual description is crucial. The names of plants should be correct, which requires advice from an agriculture specialist. Depending on extra details about context, metrics value alters up to 20%.

Images from iNaturalist of the classes most close to the target dataset show the worst results. Its top-5 accuracy is still lower than top1-accuracy of CLIP with Vision-Transformer as a backbone. The reason is that the iNaturalist dataset mostly contains plants in the wild and for example does not contain any kind of lettuce while 6 out of 17 classes from the target dataset are kinds of lettuce. This method also needs lots of time to manually choose appropriate classes.

Images retrieved by CLIP also show relatively poor results. This again may be because it retrieves images from the dataset it was trained on. This dataset does not focus primarily on the agricultural domain and some of these retrieved images were not appropriate. Some of them differed only with watermarks, some were of classes completely different from required.

3.2.4 Few-shot Classification

We evaluated results for different numbers k of images per class in the training dataset: 1, 2, 3, 5. For each k, we conducted five experiments with different dataset splits into train and test to estimate

Table 3.1: Results for zero-shot classification

Model	F1-score	top-1acc.	top-2acc.	top-3acc.	top-4acc.	top-5acc.
RN50 (add. data: iNat.)	0.12	0.14	0.26	0.34	0.41	0.50
RN50 (add. data: CLIP)	0.23	0.23	0.33	0.44	0.52	0.60
RN50 (CLIP)	0.27	0.35	0.46	0.53	0.60	0.67
ViT-B/16 (CLIP)	**0.48**	**0.51**	**0.70**	**0.78**	**0.84**	**0.88**

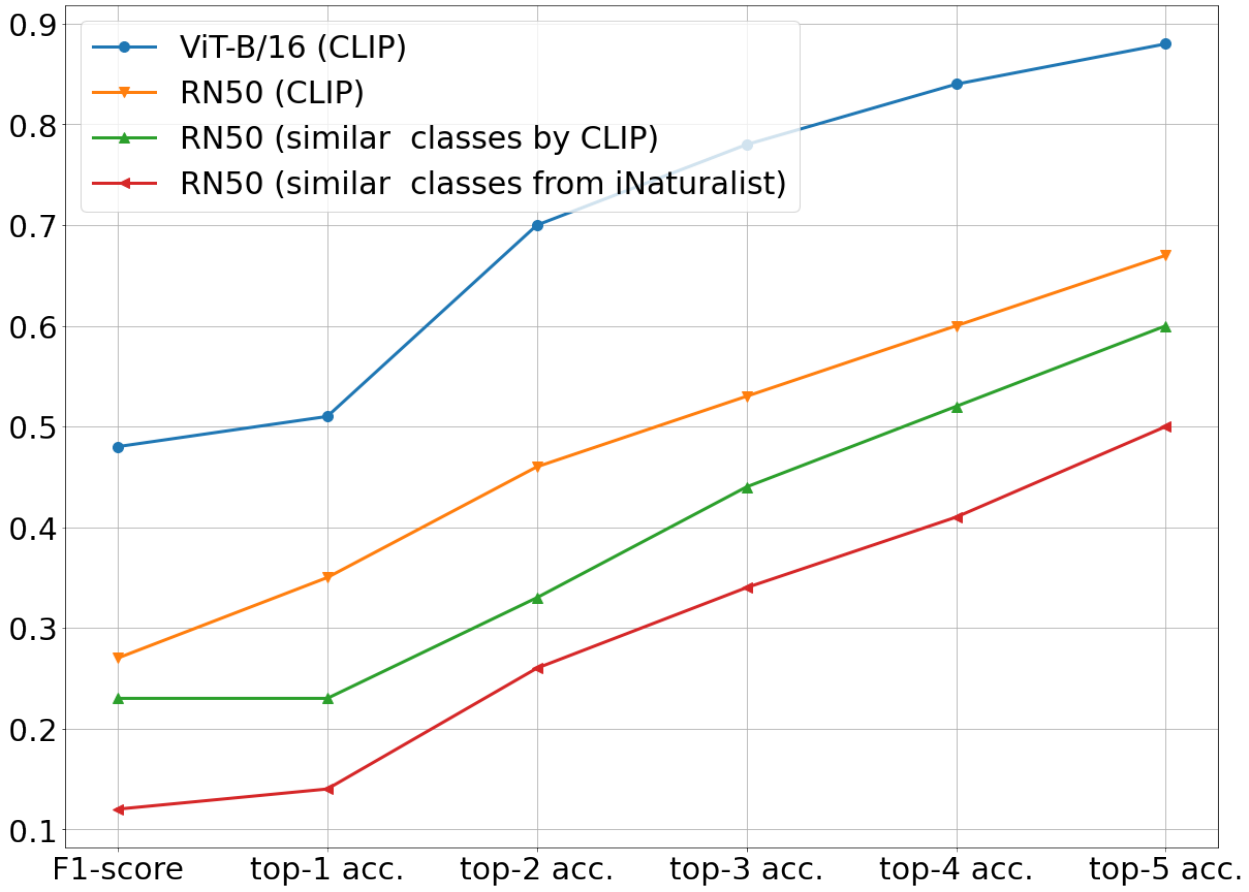

Figure 3.2: Metrics for classification in zero-shot scenario.

the mean and standard deviation of the metric. As a backbone, we used the following pretrained models:

- ResNet50 pretrained on ImageNet as the baseline;
- ResNet50 pretrained on full iNaturalist;
- CLIP with ResNet50 as the backbone;
- CLIP with Vision Transformer-B/16 as the backbone.

For all these models, we froze all parameters and trained only the last fully-connected layer with an output dimension equal to the number of classes. We used batch-size of 64, $Adam$ optimizer starting with learning rate $5 \cdot 10^{-4}$, $ReduceLROnPlateau$ scheduler and trained for 50 epochs.

Also, we further fine-tuned models from experiments for zero-shot learning:

- ResNet50 pretrained on the part of iNaturalist with images of plants of the same family as plants in the target dataset
- ResNet50 pretrained on images of same classes as in target dataset retrieved with CLIP.

We trained these models using the same settings for few-shot learning except for we used warmup for 5 epochs from 0 to $1 \cdot 10^{-4}$ and $StepLR$ scheduler with reducing step every 15 epochs by 5 times.

Results and discussion.

Results of experiments for few-shot learning are presented in Table 3.2 and in Fig. 3.3.

The iNaturalist dataset consists of images of plants taken in the wild. Moreover, it does not include most of the classes from the target dataset. As a result, it differs strongly from our target dataset. However, pretraining on it shows the best results for all k from 1 to 5 among other pretraining strategies.

The reason why such a successful model as CLIP shows relatively poor results might be that it was trained on the dataset that did not contain various enough samples from the agricultural domain; thus, it produces feature representations that lack information to distinguish plant varieties. The confusion matrix of CLIP with Vision Transformer as a backbone is presented in Fig. 3.4. Label ”5” is most often wrongly classified. The model predicts label ’4’ for five out of eight samples test samples of label ’5’. If we look at images of these two labels, we see that these are two different but still similar kinds of lettuce. Thus, it is better to train on domain-specific datasets than to use a general-case large state-of-the-art model for a fine-grained dataset.

Table 3.2: F1-scores of k-shot classification with $k \in \{1, 3, 5\}$.

Model	k=1	k=3	k=5
RN50 (ImageNet)	0.53 ± 0.03	0.75 ± 0.04	0.83 ± 0.02
RN50 (iNaturalist)	$\mathbf{0.61 \pm 0.06}$	$\mathbf{0.82 \pm 0.06}$	$\mathbf{0.88 \pm 0.03}$
RN50 (similar classes from iNaturalist)	0.55 ± 0.09	0.77 ± 0.05	0.87 ± 0.02
RN50 (similar classes by CLIP)	0.51 ± 0.05	0.72 ± 0.03	0.84 ± 0.02
RN50 (CLIP)	0.53 ± 0.06	0.65 ± 0.06	0.77 ± 0.05
ViT-B/16 (CLIP)	0.52 ± 0.10	0.75 ± 0.04	0.84 ± 0.03

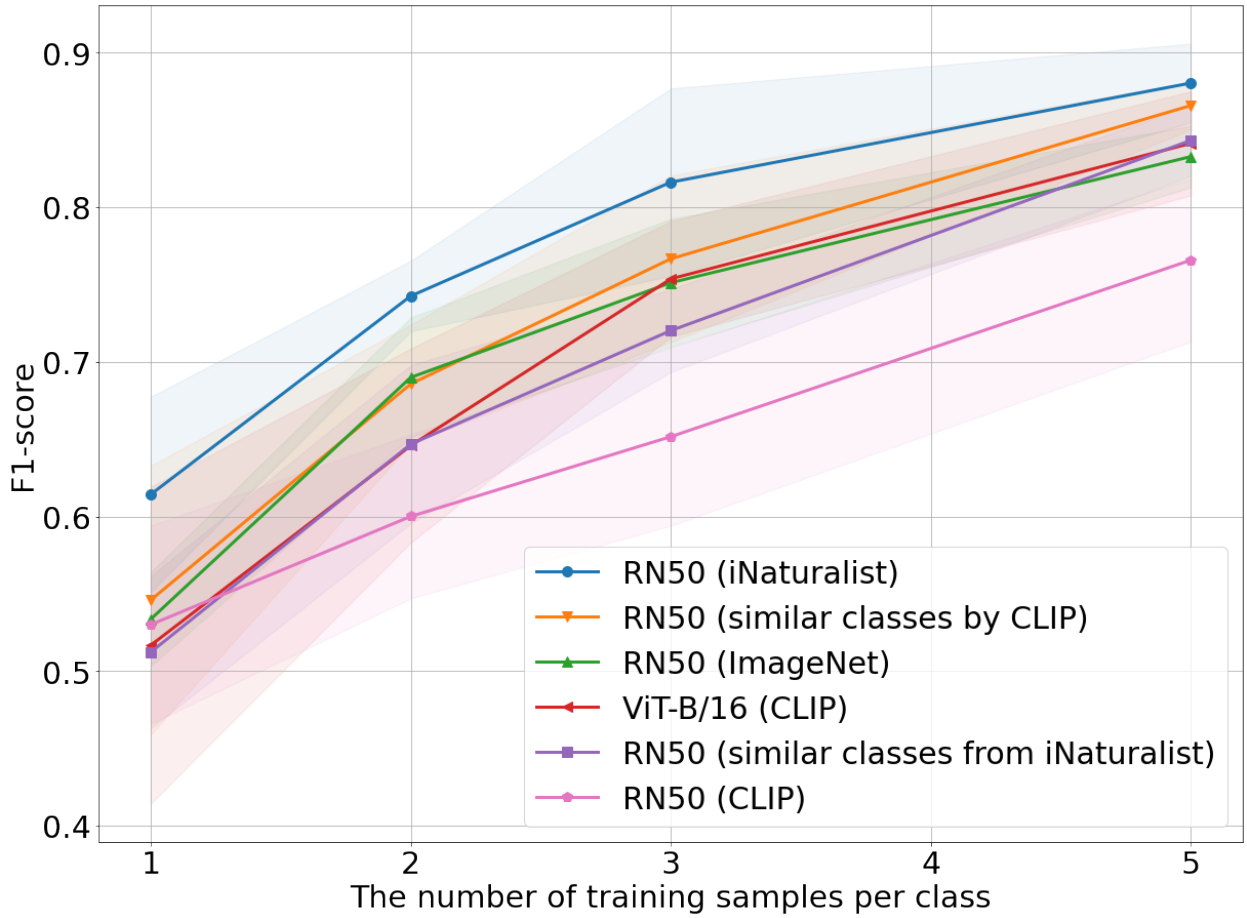

Figure 3.3: F1-scores of classification for k-shot scenario depending on number of training examples per class.

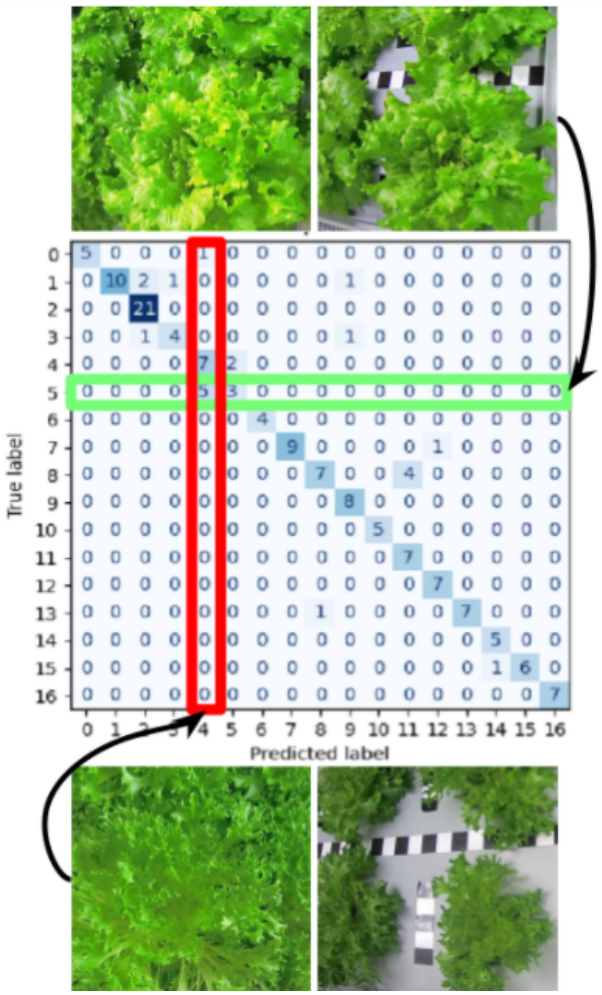

Figure 3.4: Confusion matrix of predictions by 5-shot CLIP ViT/16. The model confuses loose-leaf lettuce (label ”4”) and mizuna lettuce (label ”5”).

Chapter 4

Out-of-distribution (OOD) detection with fruit dataset

4.1 Methods

The idea of the VOS method is that we sample outliers right in the embedding space with no need for additional data with real outliers [10]. We assume that embeddings of objects of each class have multivariate Gaussian distribution [29]

$$p_\theta(h(\mathbf{x}, \mathbf{b}) \mid y = k) = \mathcal{N}\left(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}\right), \tag{4.1}$$

where $h(\mathbf{x}, \mathbf{b})$ is the embedding of the bounding box $\mathbf{b}$ from the image $\mathbf{x}$, $\boldsymbol{\mu}_k$ is the mean of distribution for class k, $k \in \{1, \dots K\}$, K is the number of in-distribution classes, $\boldsymbol{\Sigma}$ is a tied covariance matrix.

On each step, we update the queue of Q embeddings for each class and estimate mean and covariance

$$\widehat{\boldsymbol{\mu}}_k = \frac{1}{N_k} \sum_{i:y_i=k} h\left(\mathbf{x}_i, \mathbf{b}_i\right), \tag{4.2}$$

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{k} \sum_{i:y_i=k} \left(h\left(\mathbf{x}_i, \mathbf{b}_i\right) - \widehat{\boldsymbol{\mu}}_k\right) \left(h\left(\mathbf{x}_i, \mathbf{b}_i\right) - \widehat{\boldsymbol{\mu}}_k\right)^\top. \tag{4.3}$$

After that we synthesize outliers $\mathbf{v_k}$, $k \in \{1, \dots K\}$ of these estimated distributions. For that, we sample 1000 vectors from each distribution and choose vectors with the smallest values of probability density functions

$$\boldsymbol{p_k}(\mathbf{v_k}) = \frac{1}{(2\pi)^{m/2} |\hat{\boldsymbol{\Sigma}}|^{1/2}} \exp\left(-\frac{1}{2} \left(\mathbf{v}_k - \widehat{\mu}_k\right)^\top \hat{\boldsymbol{\Sigma}}^{-1} \left(\mathbf{v}_k - \widehat{\mu}_k\right)\right). \tag{4.4}$$

Finally, we compute logits for both real embeddings and synthesized outliers-embeddings

$$f(h(\mathbf{x}, \mathbf{b}); \theta) = W_{\text{cls}}^{\top}(h(\mathbf{x}, \mathbf{b}), \tag{4.5}$$

$$f(\mathbf{v}; \theta) = W_{\text{cls}}^{\top}\mathbf{v}. \tag{4.6}$$

To compute loss and to decide whether the input belongs to the in-distribution class or not, we use the log partition function F

$$F((\mathbf{x}, \mathbf{b}); \theta) := \log \sum_{k=1}^{K} e^{f_k((\mathbf{x},\mathbf{b});\theta)}. \tag{4.7}$$

F is proportional to $\log(p(x))$ [10].Its negative value $E = -F$ also known as free energy was shown to be a suitable measurement of uncertainty for OOD detection in [34].

To optimize the performance, the authors suggest to use additional loss for uncertainty

$$\mathcal{L}_{\text{uncertainty}} = \mathbb{E}_{\mathbf{v}\sim\mathcal{V}}\mathbb{I}\{F(\mathbf{v}; \theta) > 0\} + \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\mathbb{I}\{F(\mathbf{x}; \theta) \leq 0\}. \tag{4.8}$$

For the uncertainty loss to be differentiable, the loss above is replaced with binary sigmoid loss

$$\mathcal{L}_{\text{uncertainty}} = \mathbb{E}_{\mathbf{v}\sim\mathcal{V}}\left[-\log \frac{1}{1 + \exp^{-F(\mathbf{v};\theta)}}\right] + \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[-\log \frac{\exp^{-F(\mathbf{x};\theta)}}{1 + \exp^{-F(\mathbf{x};\theta)}}\right]. \tag{4.9}$$

The final optimization task is formulated in the following way:

$$\min_{\theta} \mathbb{E}_{(\mathbf{x},\mathbf{b},y)\sim\mathcal{D}}\left[\mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{loc}}\right] + \beta \cdot \mathcal{L}_{\text{uncertainty}}, \tag{4.10}$$

where $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{loc}}$ are classification and bounding boxes regression losses; $\beta = 1$ in our experiments.

During inference, we compute F with equation 4.7 for each predicted bounding box. The OOD/ID box classification is defined by the threshold γ, which is chosen so that the majority of ID boxes are correctly classified as ID

$$G\left(\mathbf{x}^*, \mathbf{b}^*\right) = \begin{cases} \text{ID}, & \text{if } F \geq \gamma, \\ \text{OOD}, & \text{if } F < \gamma. \end{cases} \tag{4.11}$$

4.2 Experiments and results

We demonstrate the performance of the method in the use case of fruit detection. The motivation and practical value are described at the end of chapter 1.

4.2.1 Data

Datasets from Open Images and COCO.

We collected relevant images from Open Images [26] and COCO [32] datasets. There are 15 fruit classes in Open Images: *banana, apple, orange, strawberry, tomato, lemon, pear, grape, watermelon, pineapple, pomegranate, grapefruit, peach, mango, common fig* and *cantaloupe*. We filtered images with group bounding boxes for fruit classes for the training stage and for the mAP calculation on ID dataset. However, we leave such images for computing OOD metrics to increase number of images and, thus, to obtain more realistic evaluation. The classes are sorted in decreasing order depending on the number of bounding boxes. We chose the first seven classes to be ID classes and the rest classes we consider to be OOD.

There are mainly 3 fruits classes in COCO: *apple, orange* and *banana*. However, we use LVIS annotation which contains more detailed bounding boxes and more fruit classes [14]. We consider the same classes as for Open Images to be ID and the rest - OOD. However, as there are few images with OOD classes, we just remove OOD images from ID test dataset and do not use them as a separate OOD dataset.

OOD similar and OOD different classes.

First of all, we wanted to look at how close to each other embeddings of objects of different classes are. For that, we trained the baseline model on all classes and ran the images through the trained model to get descriptors of ground truth bounding boxes. Then, we visualized the descriptors with Umap [36] in Fig. 4.1. We can see that some classes are separable enough such as *banana, strawberry, tomato, watermelon, pineapple, common fig*. However, *orange, lemon, grapefruit* form one rather blended group; *apple, peach, grape, mango* are strongly mixed too. That is why it seems reasonable to separate OOD classes into two groups: similar to some ID classes and different from all of them. Then we can calculate metrics for each case.

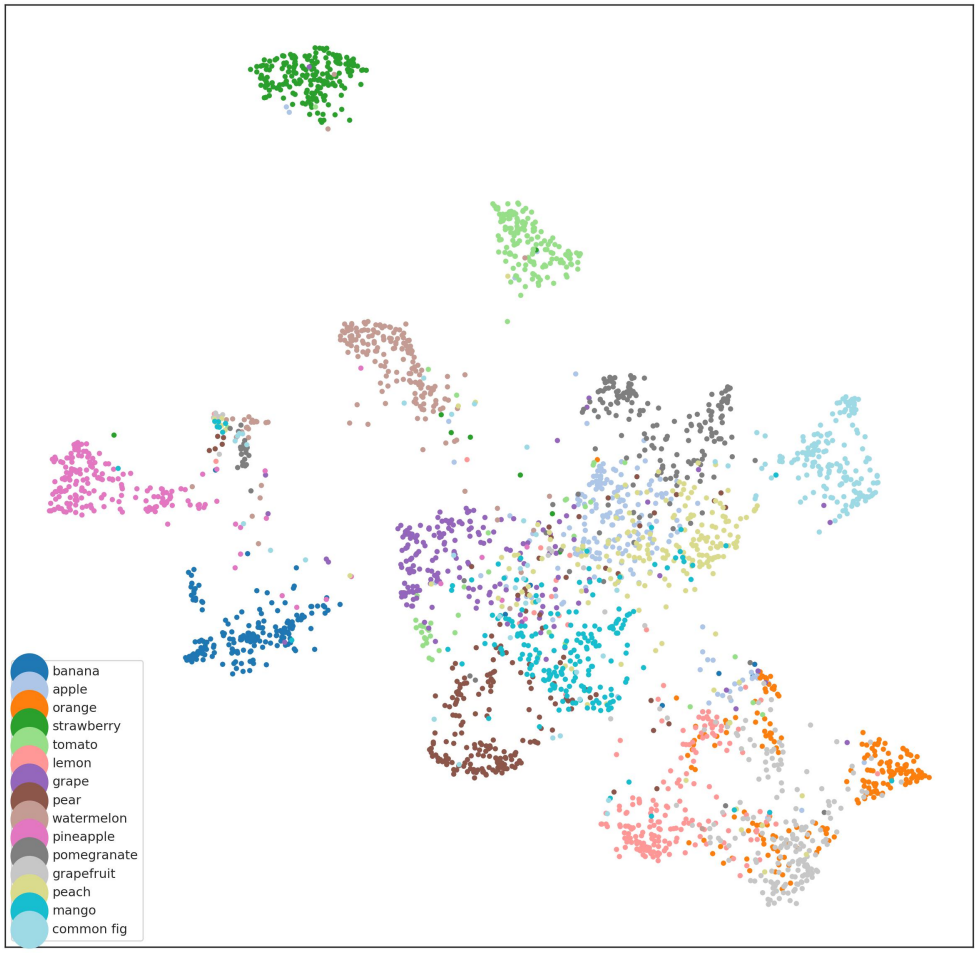

Figure 4.1: Umap visualisation of all classes.

classname	openim train	openim test	coco train	coco test	deep fruits	type
orange	4217	139	11477	1116	274	ID
banana	714	11	39299	4619	-	
apple	2023	87	15329	1701	359	
strawberry	6251	253	3553	484	348	
tomato	4263	315	10874	1312	-	
lemon	1357	127	1886	235	-	
pear	678	10	988	148	-	
grape	198	70	4269	216	-	OOD (sim)
grapefruit	311	14	-	-	-	
peach	289	35	889	155	-	
mango	211	11	-	-	401	
watermelon	611	26	704	52	-	OOD (diff)
pineapple	532	17	1530	169	-	
pomegranate	341	16	-	-	-	
common fig	208	22	-	-	-	
cantaloupe	108	13	-	-	-	
avocado	-	-	986	123	178	
rockmelon	-	-	-	-	137	

Table 4.1: Number of bounding boxes and its type for each fruits class fruits datasets.

Dataset from target domain.

As an example dataset from the target domain we took publicly available *Deep Fruits* dataset [45]. It consists from 587 RGB images with bounding boxes. According to training data described above, *apple, orange* and *strawberry* are ID classes and *mango, avocado* and *rockmelon* are OOD classes.

Test setup.

To investigate how effective the method is depending on how similar ID and OOD are, we organized four different testing setups:

- ID classes from Deep Fruits vs OOD classes from Deep Fruits;
- ID fruit classes from Open Images vs OOD-similar fruit classes from Open Images;
- ID fruit classes from Open Images vs OOD-different fruit classes from Open Images;
- ID fruit classes from COCO as ID dataset vs all other COCO classes as OOD dataset.

The final statistics on datasets is presented in Table 4.1 and Table 4.2.

base dataset	type	train	val	test
Deep Fruits	ID	-	-	163
	OOD	-	-	166
Open Images	ID	3500	617	346
	OOD (sim)	-	-	166
	OOD (diff)	-	-	144
COCO	ID	2218	389	173
	OOD (not fruits)	-	-	4512

Table 4.2: Number of images in used datasets.

4.2.2 Evaluation

To evaluate the performance of the methods, first, we need to define a threshold for the objectness scores. For that, we run a test on the validation subset of the Open Images ID dataset and choose a score, that maximizes the average class F1-score. Then, we run tests on ID and OOD test datasets and collect predicted bounding boxes with objectness scores higher than the found threshold. Boxes for ID dataset we consider to be ID and boxes for OOD dataset we consider to be OOD. Finally, we compute energy scores for ID and OOD bounding boxes, build ROC and Precision-Recall curves based on the energy scores for this binary classification problem and compute areas under these curves - AUROC and AUPR. Also, we report the mAP metric on each ID dataset.

4.2.3 Experiment details

Our code is based on Detectron2 library [56] and VOS [10]. We use Faster-RCNN [44] with FPN [31] and ResNet50 [16] as a backbone. As initialization for the backbone, we take weights from the model pretrained on the ImageNet classification task. The images during train and test are resized so that the maximum side equals 640. We train for 9000 iterations which are approximately 10 epochs on one GPU card. We use StepLR scheduler with factor $\gamma = 0.1$ at 6000 and 8000 steps starting with learning rate = 0.01. All the other settings are the same as in the base Detectron2 configuration for Faster-RCNN with FPN. For experiments with the VOS method, in addition to the hyperparameters described above, we use the original hyperparameters proposed in the paper [10] starting from iteration 6000 and sample number $Q = 500$.

4.2.4 Results and discussion

Comparative results of the models trained with and without VOS method for test fruits datasets made from COCO, Open Images, and Deep Fruits images in Table 4.3 and in Fig. 4.2.

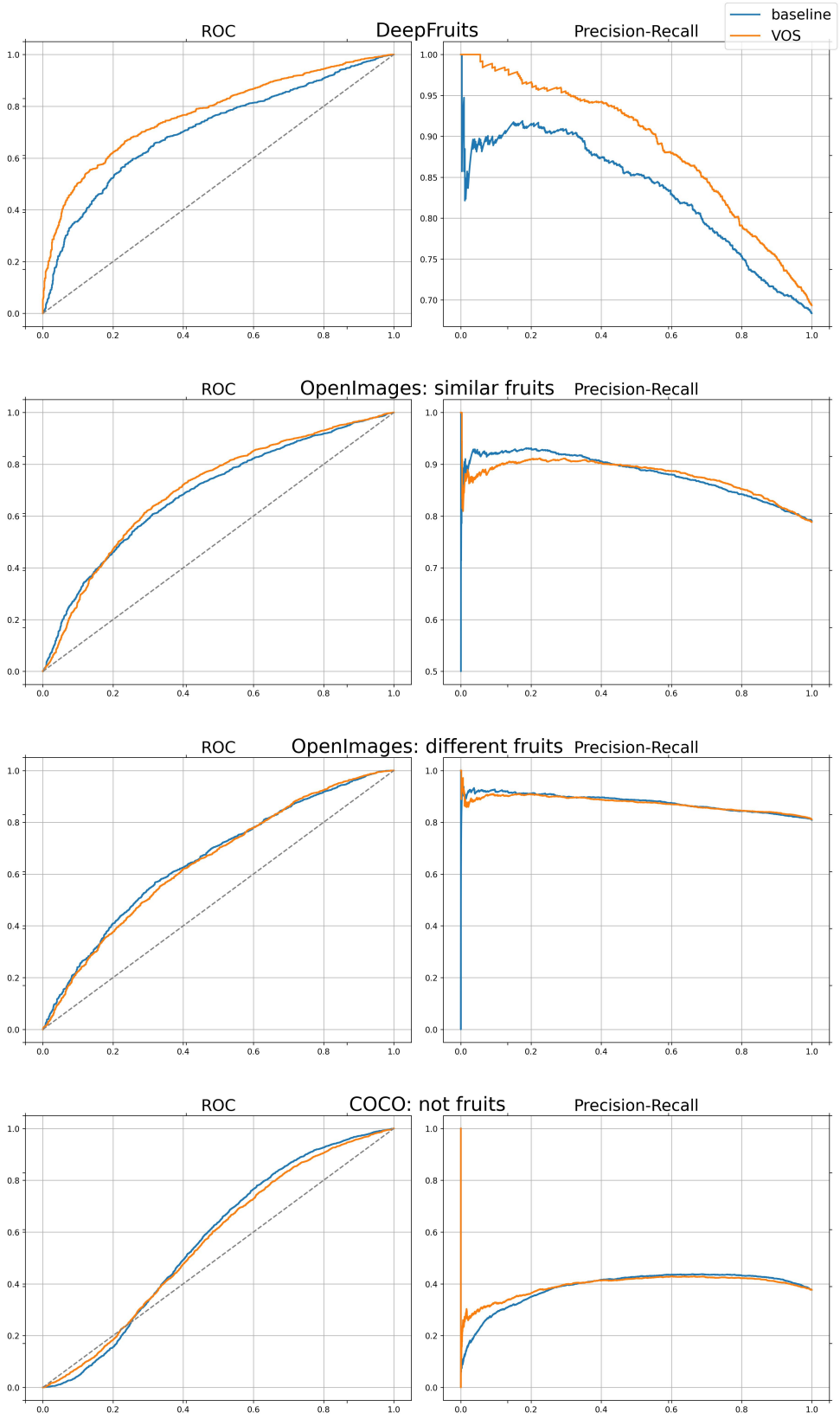

Figure 4.2: ROC and Precision-Recall curves for Deep Fruits, Open Images and COCO datasets by models with and without VOS method.

base dataset	mAP_ID (w/o)/with VOS↑, %	OOD dataset	VOS	AUROC↑, %	AUPRC↑, %
DeepFruits	**78.07 / 78.07**	fruits_ood	w/o	70.76	83.22
			with	**77.20**	**88.83**
Open Images	57.30 / **58.14**	fruits_ood_sim	w/o	68.95	**88.18**
			with	**70.18**	87.79
		fruits_ood_diff	w/o	**65.57**	**87.79**
			with	64.72	87.36
COCO	**48.17** / 47.12	no_fruits	w/o	**56.96**	38.64
			with	56.28	**39.27**

Table 4.3: Comparative results of the models trained with and without VOS method for test fruits datasets made from COCO, Open Images and Deep Fruits images.

Deep Fruits

The example results images for ID and OOD datasets are shown in Fig. 4.3 and Fig. 4.4.

We can see that in hard cases for ID classes such as green strawberry and green oranges the baseline model assigns a wrong ID class with a high score, while the model trained with the VOS method, names such objects ”OOD”. This is a more rational approach because we want the answers to be true if the scores are high. If the model considers an object to be OOD, we can run an additional classifier on a detected crop for a more precise class. This additional classifier can be trained on a wider set of classes and include OOD classes as there are more available datasets for classification task partly because labeling for classification task is quicker and cheaper to make than labeling for object detection.

For the example pictures with OOD classes, the situation is similar. The baseline model detects almost all OOD objects and assigns an ID class with a high score. It confuses OOD class with ID classes when the OOD object has some similar features like shape (OOD ’mango’ and ID ’pear’ and ’lemon’), texture (OOD ’avocado’ and ID ’strawberry’), and color (OOD ’rockmelon’ and ID ’orange’). However, the model trained with VOS detects such objects correctly as OOD.

However, we see that the AUROC score is not as good as AUROC scores reported by the authors. This is probably because the ID and OOD objects are from the same domain and are similar to each other.

COCO and Open Images

We see in Table 4.3 that for the test datasets made from COCO and Open Images the benefit from the VOS method is not obvious. This might be because the evaluation method is suitable only for cases when all detected objects are true positive on the ID dataset and all detected objects are false

positive on the OOD dataset.

These conditions are fulfilled for the Deep Fruits dataset. There are no ambiguous OOD objects which the model might detect as positive ones; so, we have no OOD detections in the ID dataset. The annotation is clean so that images with OOD objects do not contain ID objects because there can be only one kind of fruit on a tree. However, the situation is different for COCO and Open Images datasets. Let's look at the examples from the COCO dataset shown in Fig. 4.5.

The examples in the left column illustrate the problem of false-positive detection in the ID dataset. Small objects are harder to detect correctly and the AP metric is significantly lower for such objects [49]. Thus, there might be more detected OOD objects which we consider to be ID for such datasets. Another possible reason for a large number of false-positive detections is that our train dataset is not varied enough for the model to obtain common-sense opposite to datasets like Pascal VOC [11] and BDD [57] that are used in the original paper [10]. In such a case, the assumption that all objects from the ID dataset are ID becomes wrong and calculated metrics are not indicative of the effectiveness of the VOS method. The examples in the right column illustrate the problem of the appearance of OOD objects in the ID dataset. Although we use LVIS annotation which is much more detailed and contains more classes than the original COCO annotation, there are still ID objects without annotation which we cannot filter automatically using annotation from the OOD dataset. As a result, the model detects these objects and we consider them to be OOD as they are from the OOD dataset which is incorrect. The only solution for this problem seems to manually examine the OOD dataset not to contain ID objects.

To overcome these problems, we suggest to consider ID objects only predicted bounding boxes which overlap significantly with ground truth bounding boxes of any ID class. To estimate overlap it is better to use IoA metric:

$$IoA(bbox_{pred}, bbox_{gt}) = \frac{\text{Area of overlap}}{\text{Area of } bbox_{pred}}, \tag{4.12}$$

because sometimes ground truth bounding boxes are for the objects groups for simplicity of labeling. The model might detect each of these objects separately and we should consider these detections as ID. If the annotation is not full, we should label bounding boxes for all ID classes or filter images that contain not missed ID objects.

There is another possible reason why metrics on COCO and Open Images are worse than metrics on DeepFruits. Samples of some fruit classes are very dissimilar in COCO and Open Images: for example an orange on a tree vs a pealed chopped orange in a fruit salad; a more rare form of ID class might be assumed to be OOD. Fruits in DeepFruits dataset are all whole and

look similar to their standard forms. This is different for datasets used in the original paper where samples of such classes as *cow, zebra, airplane* etc., are more or less similar to each other. That is why, if objects of the same class have very different forms, it might be not effective to assume the distribution of their embeddings to be Gaussian. Probably, it is closer to a Gaussian Mixture.

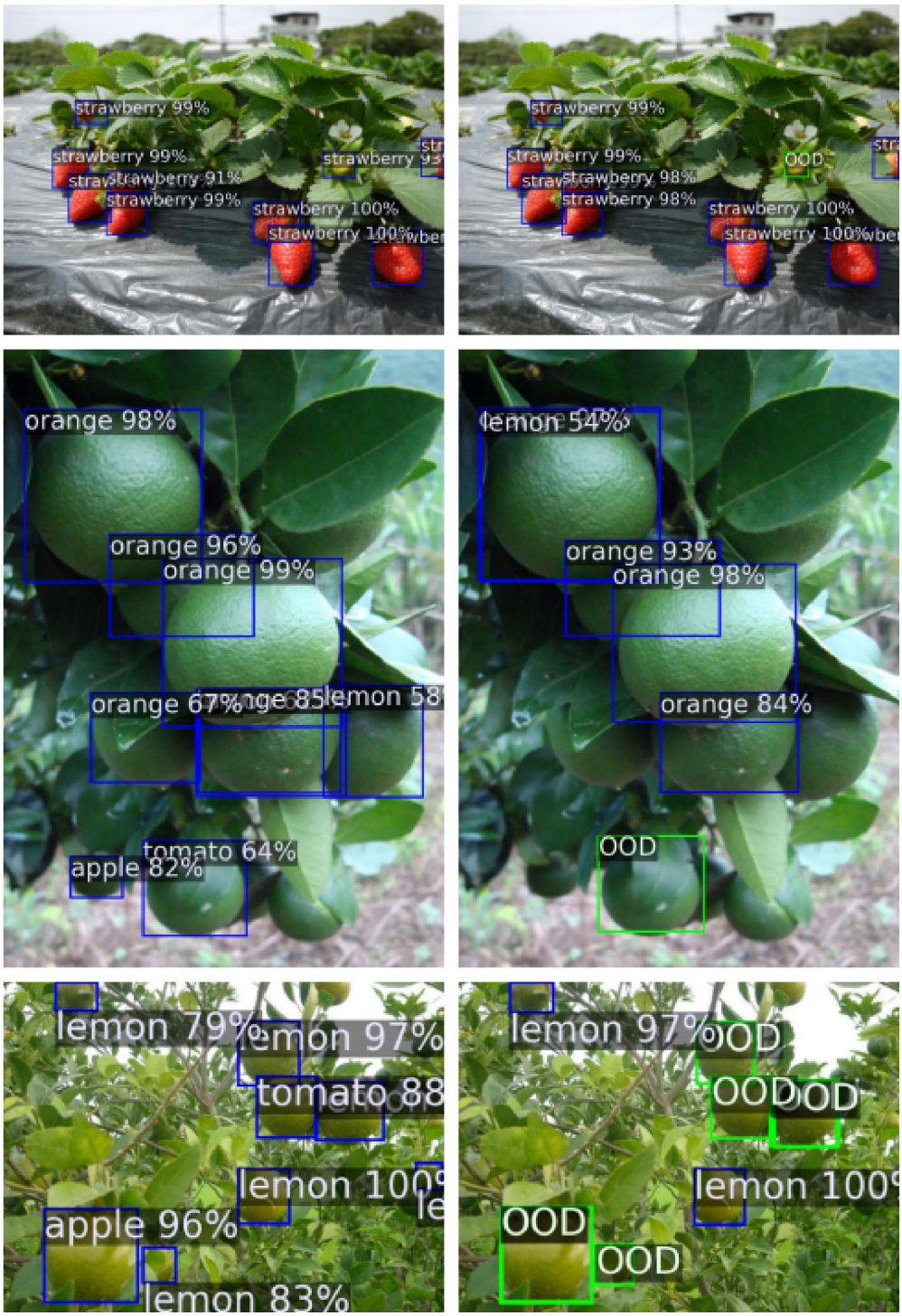

Figure 4.3: Dataset of ID classes. Left column: results by baseline model; right column: results by the VOS method.

Figure 4.4: Dataset of OOD classes. Left column: results by baseline model; right column: results by the VOS method.

Figure 4.5: Examples of detections that become problematic for calculating FPR on COCO dataset. The pictures on the left are from ID dataset; thus all detections, including false positive ones, are considered to be ID. The pictures on the right are from OOD dataset; thus all detections, including true positive ones, are considered to be OOD.

Chapter 5

Conclusions

In this work, we were solving the problem of plant phenotyping for classification and object detection cases in a data-poor environment. We aimed to compare different strategies, find their pros and cons, and come up with the most suitable ones depending on the target conditions.

5.1 Classification

5.1.1 Zero-shot

If there are no images from the target domain during the training stage and no additional features such as text description during the test stage Contrastive Language-Image Pretraining approach works best, especially with a sophisticated architecture such as Vision Transformer. Datasets manually formed from the images of the same classes taken from large open-source datasets such as iNaturalist are less effective if the target domain differs significantly. Moreover, it is the most time-consuming method. Datasets made from images retrieved by the CLIP model could be helpful especially if the pretraining dataset is increased and includes more pairs of images and their captions from the agricultural domain. The benefit of this approach is that it is possible to describe the target domain and to get images not only of the same classes but also in the environment similar to the target one such as 'greenhouse', 'open air', 'night' etc. In addition, it is easier than manually looking for relevant classes in open datasets.

5.1.2 Few-shot

If the number of images per class in a training dataset is small, using a dataset closer to the agricultural domain such as iNaturalist is a better alternative to ImageNet for pretraining. If you have computational and memory resources, it is worth trying a more diverse dataset such as a newer and bigger version of iNaturalist or a dataset formed from several agricultural datasets including Plantnet [12] besides iNaturalist. Moreover, as we saw for zero-shot CLIP a more sophisticated backbone than ResNet50 might give a significant improvement for fine-grained datasets. Finally, it

could be beneficial to try other pretraining strategies besides supervised learning with cross-entropy. For example, methods of self-supervised learning such as approaches based on Contrastive Learning [5], [4] or Masked Autoencoders [15].

5.2 Object detection

If there are no data for out-of-distribution classes, VOS is a promising method to reduce the number of false-positive detections especially if objects are similar to each other within each in-distribution class. If objects within some in-distribution classes differ strongly, the assumption that embeddings have class-conditional multivariate Gaussian distribution might be incorrect and the original method might work not that good. It is crucial that you have a complete and careful annotation for objects of in-distribution classes; so, if you use images from COCO, it is better to use LVIS annotation. Moreover, the original method to define threshold is not correct if there are many false-positive detections on images with in-distribution objects. This becomes especially relevant when the objects are small and the training dataset is not various enough for the model to obtain common sense. To overcome this problem, it might be better to take as in-distribution objects only predicted bounding boxes that overlap significantly with ground truth boxes of any class and/or add to the training dataset all images from a large open dataset like COCO without ground-truth bounding boxes if they do not contain in-distribution classes.

Publications

Liliya Lemikhova, Sergey Nesteruk and Andrey Somov. Transfer Learning for Few-Shot Plants Recognition: Antarctic Station Greenhouse Use-Case. In *International Symposium on Industrial Electronics*, Anchorage, Alaska, USA, June 2022.

Acknowledgement

I would like to thank German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt e.V.) for sharing data for this research. Also, I would like to express my gratitude to Sergey Nesteruk for his support and advice during the work.

Bibliography

[1] David Argüeso, Artzai Picon, Unai Irusta, Alfonso Medela, Miguel G San-Emeterio, Arantza Bereciartua, and Aitor Alvarez-Gila. Few-shot learning approach for plant disease classification using images taken in the field. *Computers and Electronics in Agriculture*, 175:105542, 2020.

[2] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016.

[3] Dupont C., Hermenier F., Schulze T., Basmadjian R., Somov A., and Giuliani G. Plug4green: A flexible energy-aware vm manager to fit data centre particularities. *Ad Hoc Networks*, 25(PB):505 – 519, 2015.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[6] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4109–4118, 2018.

[7] 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.

[8] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[9] 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.

[10] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis. *Proceedings of the International Conference on Learning Representations*, 2022.

[11] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[12] Camille Garcin, Alexis Joly, Pierre Bonnet, Jean-Christophe Lombardo, Antoine Affouard, Mathias Chouet, Maximilien Servajean, Joseph Salmon, and Titouan Lorieul. Pl@ ntnet-300k: a plant image dataset with high label ambiguity and a long-tailed distribution. In *NeurIPS 2021-35th Conference on Neural Information Processing Systems*, 2021.

[13] Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. *arXiv preprint arXiv:1810.13243*, 2018.

[14] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[16] 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.

[17] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[18] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.

[19] Mohammad Hossin and Md Nasir Sulaiman. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 5(2):1, 2015.

[20] David Hughes, Marcel Salathé, et al. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*, 2015.

[21] Svetlana Illarionova, Sergey Nesteruk, Dmitrii Shadrin, Vladimir Ignatiev, Mariia Pukalchik, and Ivan Oseledets. Object-based augmentation for building semantic segmentation: Ventura and santa rosa case study. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1659–1668, 2021, 2021.

[22] Svetlana Illarionova , Sergey Nesteruk , Dmitrii Shadrin, Vladimir Ignatiev , Maria Pukalchik , and Ivan Oseledets. Mixchannel: Advanced augmentation for multispectral satellite images. *Remote Sensing*, 13(11), 2021.

[23] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5830–5840, 2021.

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

[25] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[26] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

[27] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017.

[28] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

[29] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In S. Bengio, H. Wallach,

H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[30] Yang Li and Jiachen Yang. Few-shot cotton pest recognition and terminal realization. *Computers and Electronics in Agriculture*, 169:105240, 2020.

[31] 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.

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[33] Lars Lindner, Oleg Sergiyenko, Moises Rivas-López, Daniel Hernández-Balbuena, Wendy Flores-Fuentes, Julio C Rodríguez-Quiñonez, Fabian N Murrieta-Rico, Mykhailo Ivanov, Vera Tyrsa, and Luis C Básaca-Preciado. Exact laser beam positioning for measurement of vegetation vitality. *Industrial Robot: An International Journal*, 2017.

[34] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020.

[35] Yuzhen Lu and Sierra Young. A survey of public datasets for computer vision tasks in precision agriculture. *Computers and Electronics in Agriculture*, 178:105760, 2020.

[36] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[37] Sina Mohseni, Mandar Pitale, JBS Yadawa, and Zhangyang Wang. Self-supervised learning for generalizable out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5216–5223, 2020.

[38] Sergey Nesteruk, Svetlana Illarionova, Timur Akhtyamov, Dmitrii Shadrin, Andrey Somov, Mariia Pukalchik, and Ivan Oseledets. Xtremeaugment: Getting more from your data through combination of image collection and image augmentation. *IEEE Access*, 10:24010–24028, 2022.

[39] Sergey Nesteruk, Dmitrii Shadrin, Vladislav Kovalenko, Antonio Rodríguez-Sánchez, and Andrey Somov. Plant growth prediction through intelligent embedded sensing. In *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*, pages 411–416, 2020, 2020.

[40] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[41] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[42] Roop Pahuja, H.K. Verma, and Moin Uddin. A wireless sensor network for greenhouse climate control. *IEEE Pervasive Computing*, 12(2):49–58, 2013.

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[44] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[45] Inkyu Sa, Zongyuan Ge, Feras Dayoub, Ben Upcroft, Tristan Perez, and Chris McCool. Deepfruits: A fruit detection system using deep neural networks. *sensors*, 16(8):1222, 2016.

[46] Oleg Sergiyenko, Vera Tyrsa, Wendy Flores-Fuentes, Julio Rodriguez-Quiñonez, and Paolo Mercorelli. Machine vision sensors. *Journal of Sensors*, 2018:1–2, 2018.

[47] Oleg Yu Sergiyenko and Vera V. Tyrsa. 3d optical machine vision sensors with intelligent data management for robotic swarm navigation improvement. *IEEE Sensors Journal*, 21(10):11262–11274, 2021.

[48] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[49] Kang Tong, Yiquan Wu, and Fei Zhou. Recent advances in small object detection based on deep learning: A review. *Image and Vision Computing*, 97:103910, 2020.

[50] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. From ImageNet to image classification: Contextualizing progress on benchmarks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference*

on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, pages 9625–9635. PMLR, 13–18 Jul 2020.

[51] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 595–604, 2015.

[52] Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisin Mac Aodha. Benchmarking representation learning for natural world image collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12884–12893, June 2021.

[53] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[54] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

[55] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[56] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[57] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

[58] Zhongjie Yu, Lin Chen, Zhongwei Cheng, and Jiebo Luo. Transmatch: A transfer-learning scheme for semi-supervised few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12856–12864, 2020.

[59] Wei Zhao, Xuan Wang, Bozhao Qi, and Troy Runge. Ground-level mapping and navigating for agriculture based on iot and computer vision. *IEEE Access*, 8:221975–221985, 2020.