Class-Incremental Learning of Plant and Disease Detection: Growing Branches with Knowledge Distillation

Mathieu Pagé-Fortin
Laval University, Canada
mathieu.page-fortin.1@ulaval.ca

Abstract

This paper investigates the problem of class-incremental object detection for agricultural applications where a model needs to learn new plant species and diseases incrementally without forgetting the previously learned ones. We adapt two public datasets to include new categories over time, simulating a more realistic and dynamic scenario. We then compare three class-incremental learning methods that leverage different forms of knowledge distillation to mitigate catastrophic forgetting. Our experiments show that all three methods suffer from catastrophic forgetting, but the Dynamic Y-KD approach, which additionally uses a dynamic architecture that grows new branches to learn new tasks, outperforms ILOD and Faster-ILOD in most settings both on new and old classes.

These results highlight the challenges and opportunities of continual object detection for agricultural applications. In particular, we hypothesize that the large intra-class and small inter-class variability that is typical of plant images exacerbate the difficulty of learning new categories without interfering with previous knowledge. We publicly release our code to encourage future work. [1]

1. Introduction

Agriculture is a fundamental sector of the global economy, and its importance will only increase as the world's population is expected to reach 9.7 billion by 2050 [10]. With a projected 70% increase in demand for food production, precision agriculture practices are gaining increasing attention to improve productivity by acting at a more refined scale [45]. The advent of precision agriculture tools and new technologies has enabled the gathering and analysis of large quantities of data with higher spatio-temporal resolution. Among these advances, computer vision has an important role in remote and proximal sensing by providing

tools to process complex images [8, 32].

Remote sensing methods, which include the analysis of data collected from drones, planes and satellites, have many applications for plant health and growth monitoring [39], yield size and quality estimation [50], decision support systems for irrigation or soil management [16], large-scale phenotyping [25], and others. Similarly, proximal sensing methods with personal cameras or cameras mounted on vehicles can be used for insect and pest detection [30], robotic-assisted weed removal or spraying [55], and more.

For many of these applications, object detection plays a crucial role in identifying the location of target instances in images, such as individual plants, weeds or insects. To solve this task, deep learning approaches have gained widespread popularity, especially Convolution Neural Networks (CNN), due to their superior performance and ability to extract relevant features directly from image data [27].

However, deep learning approaches generally assume that the training dataset is static, such that training can be performed in one step before deployment. This scenario faces limitations in dynamic environments [22] typical of agriculture (see Fig. 1). For instance, in the context of crop health monitoring, new diseases or new weed species can emerge on farms by means of contaminated crop seeds [36] or invasive weed species [44], inducing a necessity to update existing models to recognize new categories. However, incrementing deep learning models with such new classes is still a challenging task as these models are prone to catastrophic forgetting [34]. They tend to erase previous knowledge while learning new classes.

While this problem is actively studied for image classification [9], class-incremental object detection is far less explored [35]. Furthermore, previous work has mostly been done on benchmark datasets such as COCO [28] or Pascal-VOC [13]. Very few work addressed the challenges of class-incremental learning in agriculture [3, 37, 24]. For instance, the large intra-class and small inter-class variability that is characteristic of agricultural imagery (e.g. see Fig. 2 and Fig. 3) might increase catastrophic forgetting as new knowledge interferes with previously learned features. Moreover,
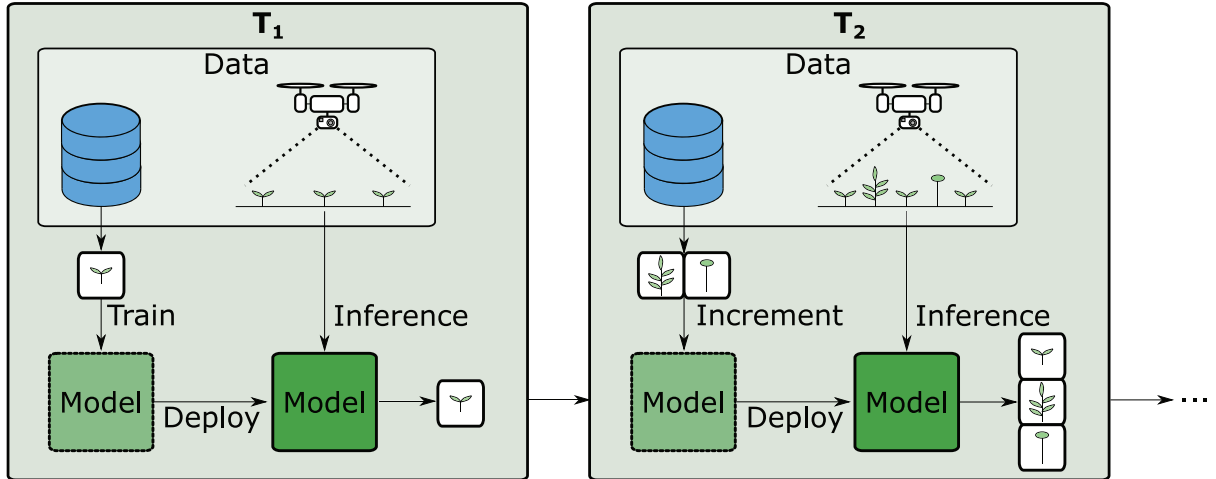
---

[1] https://github.com/DynYKD/Continual-Plant-Detection

Figure 1: Motivating example of continual learning in agricultural contexts. Initially at T_1, a detection model is trained to identify a particular species of plant and is subsequently deployed in the field for this purpose. Later at T_2, two new weed species emerge in the field. The user intends for the model to accurately detect these additional species as well. To achieve this, the model is incremented exclusively with training examples of these newly emerged weeds and redeployed to detect all three plant varieties.

as the annotation of data often requires specialists, the high cost of data acquisition can result in fewer training examples [4]. Consequently, incremental methods should exhibit high plasticity to efficiently learn from few data. On the other hand, increased plasticity generally conflicts with stability, such that forgetting can be exacerbated.

In this paper, we study class-incremental object detection on two agricultural datasets. To the best of our knowledge, we are the first to tackle such problem explicitly on agricultural imagery. We adapt two public datasets to simulate class-incremental learning scenarios: 1) the Strawberry Disease Detection Dataset [2] which is composed of 2,500 images showing seven categories of diseases, and 2) the Open Plant Phenotyping Database [21], a large dataset of plant seedlings from 47 different species. We divide the classes in distinct sets and the models have to learn them successively. We publicly release the code that adapts both datasets to class-incremental learning scenarios to encourage future research.

We compare three continual learning approaches that rely on different forms of knowledge distillation (KD) [15], a popular regularization strategy that uses the outputs of the previous model to transfer its knowledge to the learning model. Forgetting is thereby reduced as it constrains the new model from diverging significantly from its previous state. The first approach, ILOD [47], uses KD by forcing the new model to give scores and boxes of previous classes that are similar to the ones given by the previous model. The second, Faster-ILOD [40], includes additional KD losses to further guide the new model in preserving past activations while learning new categories. Finally, Dynamic Y-KD [38] is a hybrid approach that has been proposed re-

cently. It combines a new form of KD that improves the learning of new categories with a dynamic architecture that grows additional feature extractors for each new task. By using class-specific feature extractors to recognize previous categories, catastrophic forgetting has been shown to be significantly reduced.

We perform extensive experiments on the Strawberry Disease Detection Dataset and the Open Plant Phenotyping Database, and we show that while ILOD and Faster-ILOD performs comparatively, Dynamic Y-KD significantly outperforms both of them on new classes while reducing forgetting in most scenarios. However, Dynamic Y-KD incurs higher memory and computational costs at inference, which can be an issue if real-time inference is critical.

In summary, our contributions are as follows:

- We adapt two datasets of agricultural imagery to study the task of class-incremental object detection. To our knowledge, these are the first public datasets for continual learning of tasks related to agriculture. We publicly release the code to generate the same splits for reproducibility and to encourage future work.

- We compare three continual learning approaches of object detection that exploit knowledge distillation and a dynamic architecture. Notably, we show that the recent Dynamic Y-KD generally outperforms ILOD and Faster-ILOD at a cost of increased computation.

- We perform extensive experiments on various class-incremental scenarios on both datasets. Our results reveal that continual learning is especially challenging on agricultural data, most likely because of the high intra-class and low inter-class variability.

2. Related work

2.1. Object Detection in Agriculture

Object detection is a fundamental task for several applications in agriculture, including pest detection [12, 29, 53], automatic weed spraying [14, 43, 1, 19], harvesting [56], yield estimation [18, 49, 20], crop health and growth monitoring [48, 57, 7], and more.

Agricultural contexts exhibit specific challenges that complicate the application of computer vision techniques. For instance, real-time inference can be required for robotic solutions [52, 41, 59]. The objects of interest (e.g. fruits) can also be small and occluded [54], have irregular and varying shapes [31] and can appear under different lighting conditions [58]. Furthermore, acquiring and labelling data to train deep learning models is especially costly in agriculture as it can require field experts, for instance to annotate specific diseases and weed species.

Several deep learning models have been proposed to tackle these challenges. For instance, [54] compared three models with regard to their ability to handle images with large numbers of fruits and flowers of varying sizes. The authors of [31] proposed an attention mechanism that processes features from multiple resolutions and the use of Atrous Spatial Pyramid Pooling to improve the detection of occluded or small leaves. Regarding the high cost of data annotation, an active learning strategy was proposed in [4] to only label the most informative images.

However, differently from previous lines of work that aimed to improve accuracy on challenging images, this paper tackles the problem of continual learning on agricultural images.

2.2. Continual Learning

While the challenges above have been addressed in previous work, it is generally assumed that the environment is static, such that object detector models can be trained in one step and deployed afterward. Very few work considered the challenges that arise due to the dynamic nature of agricultural environments, such as the emergence of new object categories. For instance, new pests, diseases or weed species can emerge on farms by means of contaminated crop seeds [36] or invasive weed species [44]. Effective computer vision solutions should therefore be able to be incremented with new data easily. However, deep learning models are prone to catastrophic forgetting, making their adaptation to new classes while preserving previous knowledge especially challenging.

To mitigate catastrophic forgetting, continual learning solutions generally employ rehearsal [46, 51, 33], regularization losses [17, 26, 11, 38], or dynamic architectures [38, 23]. Rehearsal consists of replaying past data in training batches while learning new classes, thereby reducing the drift towards new classes. Regularization-based approaches constrain the weights of the learning model to limit knowledge loss. Finally, dynamic architectures keep specific weights for each set of classes and grow new branches to accumulate new knowledge. For a recent and detailed review on continual object detection, please refer to [35].

Previous work on continual learning in the context of agricultural applications explored the task of plant disease classification [24, 3, 37], mainly using rehearsal strategies. For instance, [24] uses a Generative adversarial network to generate abstract representations of previous classes to store them in memory and mixes them with new classes. In [3], raw images of previous tasks are stored and used after incremental learning to correct the bias towards new classes with a linear layer. Finally, [37] employs rehearsal and ensures that diversity of old data is preserved to limit forgetting.

To our knowledge, regularization strategies and dynamic architectures have not been explored for continual learning to solve tasks related to agriculture. Furthermore, the problem of class-incremental object detection has not been studied on agricultural images in previous work. Therefore, this paper contributes to continual learning in agriculture by comparing three object detection approaches using knowledge distillation and dynamic architectures.

3. Materials and Methods

In this section, we begin by formulating the problem of continual object detection in the context of agricultural applications. Next, we describe the two agricultural datasets that we used to experiment class-incremental plant and disease detection, and we summarize how we adapted them to simulate incremental learning. We then introduce the three methods that we compared, namely ILOD [47], Faster-ILOD [40] and Dynamic Y-KD [38].

3.1. Continual Object Detection in Agriculture

Continual object detection aims to increment a model $f_{\theta^{t-1}}$ that can detect objects of classes $\mathcal{C}^{0:t-1}$ to a model f_{θ^t} that can detect new classes $\mathcal{C}^t$ in addition to previous classes $\mathcal{C}^{0:t-1}$. At any step t, we are given a labelled training dataset $\mathcal{D}^t$ composed of images X^t with annotations Y^t showing examples of current classes $\mathcal{C}^t$. For object detection, the annotations Y^t are the bounding boxes and classes of the items on each image. However, the established continual learning setup [47, 6, 38] generally assumes that at any given step, training data are available for current classes exclusively, while previous and future classes are either unlabeled or unavailable. We also adopt this experimental setup in this work.

In the context of agricultural applications, such a continual learning scenario captures a situation where an object detection model $f_{\theta^{t-1}}$ has been trained and deployed

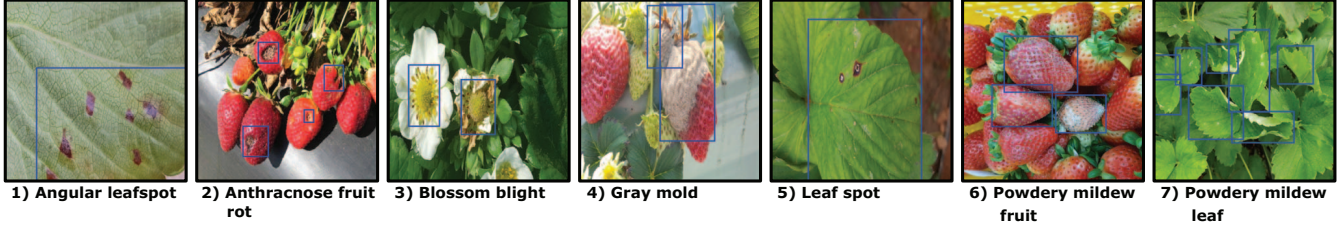| 1) Angular leafspot | 2) Anthracnose fruit rot | 3) Blossom blight | 4) Gray mold | 5) Leaf spot | 6) Powdery mildew fruit | 7) Powdery mildew leaf |

Figure 2: Strawberry Disease Detection Dataset [2]. Examples of images and annotations for the seven categories of diseases.

to solve a given task, detecting objects (e.g. weeds, diseases) from certain classes $\mathcal{C}^{0:t-1}$ in the field (see Fig. 1). However, as the environment is dynamic, new categories $\mathcal{C}^t$ might emerge at a time t, inducing the necessity to update the detection model. After a dataset $\mathcal{D}^t$ of new classes $\mathcal{C}^t$ has been acquired and labeled, the model $f_{\theta^{t-1}}$ is incremented to f_{θ^t}. This new model can then be redeployed on the field, having acquired the ability to detect new classes accurately in addition to previous ones. This scenario can repeat for multiple steps if new categories emerge in the future again.

3.2. Datasets

We adapt two public datasets to simulate incremental learning scenarios following the established experimental setups of class-incremental learning [47]. An incremental learning scenario is defined by $b - n$, where b represents the number of base categories learned initially, and n is the number of new classes added in subsequent steps until all classes from the dataset are learned.

As an example on the Strawberry Disease Detection Dataset [2], a scenario denoted by *3-2* designates that the models are trained on categories *1-3* in a first step, then they are incremented exclusively on classes *4-5*, and finally they are trained exclusively on classes *6-7* to reach the total of 7 categories of diseases. At the end, model performances are evaluated on the test set composed of all classes *1-7*.

Following the experimental setup established to adapt benchmark datasets such as Pascal-VOC, the classes are simply sorted in alphabetical order. The images from each dataset are split into train and test sets. However, at each incremental step t, we only have access to the training set $\mathcal{D}^t$.

Strawberry Disease Detection Dataset (SDDD). The SDDD [2] is composed of 2,500 images showing seven types of strawberry diseases. Figure 2 illustrates an example of each class. Notice that diseases can be observed either on leaves, flowers or fruits. All images have been acquired with mobile phones and have been resized to a resolution of 419×419. This dataset is relatively small for the task, which is a common challenge of public datasets of agricultural imagery [32]. For this reason, the authors of SDDD [2] explored various data augmentation techniques and found

that some operations could slightly improve the results over a baseline. For simplicity, in this paper we did not apply such technique but future work should explore the contribution of data augmentation in incremental learning settings.

Open Plant Phenotyping Database (OPPD). The OPPD [21] regroups 7,590 images of 47 plant species. Each image captures a grow box viewed from the top in which several plants of the same species grow, making a total of 315,041 plant entities. The images have been acquired throughout the whole life cycle of the plants, such that intra-class variability is high while the inter-class variability can be quite small, especially in early stages of development (see Fig. 3).

3.3. Methods

In this work, we compare three approaches to tackle continual object detection. We begin by summarizing the Faster R-CNN model [42] since all three approaches build on this architecture. Then, we explain ILOD [47] and Faster-ILOD [40], two methods that exclusively rely on knowledge distillation (KD) [15] to prevent catastrophic forgetting. We summarize the recent Dynamic Y-KD network [38] that uses an hybrid approach between KD and a dynamic architecture.

3.3.1 Faster R-CNN

In the context of continual learning, Faster R-CNN [42] is made of a backbone $\mathcal{F}_{\theta^t}^B$, a region proposal network (RPN) $\mathcal{F}_{\theta^t}^{RPN}$ and a box head $\mathcal{F}_{\theta^t}^{box}$. Specifically, input images X^t are fed to the backbone to give feature maps $\hat{X}^t$. These features maps are then processed by the RPN which proposes several regions of interests (RoIs) with corresponding objectness scores. The box head subsequently classifies each RoI and the coordinates of bounding boxes are regressed.

The supervised loss used to train Faster R-CNN is the sum of classification and regression losses on the RPN and the box head. Following the formulation of [5], we define the loss of Faster R-CNN as follows:

$$\mathcal{L}_{faster} = \mathcal{L}_{cls}^{RPN} + \mathcal{L}_{reg}^{RPN} + \mathcal{L}_{cls}^{box} + \mathcal{L}_{reg}^{box}, \quad (1)$$
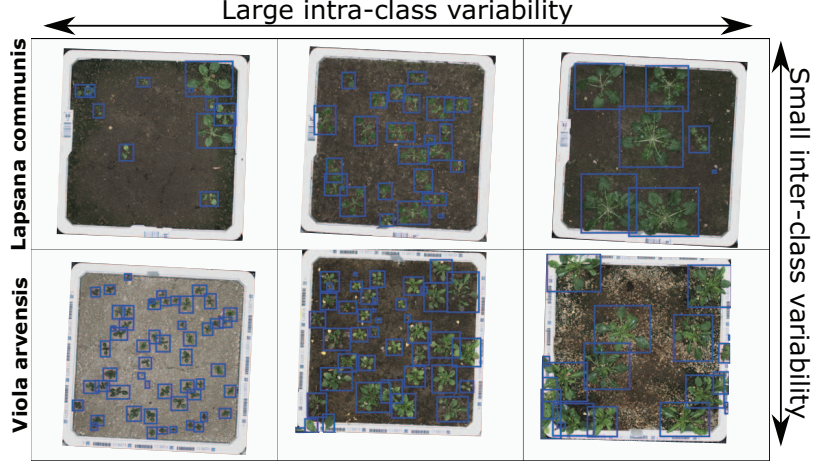
Figure 3: Open Plant Phenotyping Database [21]. Three examples of two classes and their annotations are shown to highlight the large intra-class variability and small inter-class variability that are frequent in agricultural imagery.

where $\mathcal{L}_{cls}$ are binary cross-entropy losses for classification applied on the RPN and the box head, and $\mathcal{L}_{reg}$ are regression losses of bounding boxes. For more details we refer the reader to [42].

While this training can be sufficient to learn a single detection task with a closed set of classes, catastrophic forgetting arises when facing incremental learning scenarios. For this reason, class-incremental learning techniques have been proposed to complement Faster R-CNN and address its limitations when new sets of classes are introduced.

3.3.2 ILOD

ILOD [47] is among the first deep learning methods proposed to tackle continual learning for object detection. It employs KD, a popular regularization-based strategy to reduce catastrophic forgetting. To perform incremental learning, ILOD keeps a frozen copy of the whole network from the previous step, acting as a teacher network to a student network. Training images of new classes are then given to both the new and previous models. In addition to the supervised loss, the new model is trained with an L2 loss that compares the difference between classification logits $\hat{y}$ and box coordinates r given by the previous model, as follows:

$$\mathcal{L}_{dist}^{box} = \frac{1}{N|\mathcal{C}^{0:t}|} \sum \left[(\hat{y}^t - \hat{y}^{t-1})^2 + (r^t - r^{t-1})^2 \right], \quad (2)$$

where N is the number RoIs sampled for KD. Thereby, this loss encourages the new model to keep similar outputs for previous classes, preventing the weights from diverging from their previous states.

The total loss of ILOD is then as follows:

$$\mathcal{L}_{\mathrm{ILOD}} = \mathcal{L}_{faster} + \lambda_1 \mathcal{L}_{dist}^{box}, \quad (3)$$

where λ_1 is a hyperparameter applied on the KD loss to balance the trade-off between learning and forgetting.

3.3.3 Faster-ILOD

Faster-ILOD [40] adopts a similar approach to continual object detection than ILOD, using KD as the approach to limit forgetting. However, rather than only distilling knowledge from the last layer, namely the class logits and box coordinates, the authors add a KD loss between the feature maps of the teacher and student networks, and another for the RPN. This time, an L1 loss is used for distillation of feature maps activations as follows:

$$\mathcal{L}_{dist}^{feat} = \frac{1}{HWD} \begin{cases} \sum_i ||\hat{X}_i^{t-1} - \hat{X}_i^t||_1, & \text{if } \hat{X}_i^{t-1} > \hat{X}_i^t \\ 0, & \text{otherwise} \end{cases}$$
(4)

where H, W and D are the height, width and depth of the feature maps, respectively. Notice that KD is only applied if the activation of the previous model is larger than the one of the current model. The motivation is to only distill activations that are important for previous classes.

For the RPN distillation, a similar idea of only distilling the outputs that are most likely to be important for old classes is adopted. Specifically, the distillation between the RPN of the teacher and student networks is as follows:

$$\mathcal{L}_{dist}^{RPN} = \frac{1}{N} \sum_i^N \mathbf{1}_{\left[s_i^t \geq s_i^{t-1} \right]} ||s_i^t - s_i^{t-1}|| +$$
$$\mathbf{1}_{\left[s_i^t \geq s_i^{t-1} + \tau \right]} ||\omega_i^t - \omega_i^{t-1}||, \quad (5)$$

where s_i is the objectness score of the RoI i, ω_i is its coordinates, and τ is an hyperparameter that defines a margin.

The total loss of Faster-ILOD is then as follows:

$$\mathcal{L}_{\mathrm{FILOD}} = \mathcal{L}_{\mathrm{ILOD}} + \lambda_2 \mathcal{L}_{dist}^{feat} + \lambda_3 \mathcal{L}_{dist}^{RPN}. \quad (6)$$

3.3.4 Dynamic Y-KD Network

The Dynamic Y-KD network [38] is a recent approach that adopts a hybrid strategy, mixing a novel self-distillation method with a parameter isolation approach that uses previous backbones to predict past classes while growing new branches to accommodate new tasks.

While most previous work using KD keeps a copy of the complete network, Dynamic Y-KD reduces the teacher network to the detection head. During incremental learning, the backbone is thus shared between the teacher and student network such that the teacher network head receives the same feature maps as the student.

Formally, a backbone B that is fixed at all time is connected to the student feature extractor F_θ^t to produce feature maps $\hat{X}^t$ as follows:

$$\hat{X}^t = F^t(B(X^t)). \tag{7}$$

The same feature maps are then given to the teacher detection head H^{t-1} whose weights are frozen, as well as to the trainable student head H^t as follows:

$$\hat{Y}^{t-1} = H^{t-1}(\hat{X}^t)$$
$$\hat{Y}^t = H^t(\hat{X}^t). \tag{8}$$

KD is then applied with $\hat{Y}^{t-1}$ and $\hat{Y}^t$ according to Eq. 2 and Eq. 5. The difference resides in the usage of shared feature maps to produce both outputs. With this modification to the general architecture of KD, the teacher network is made more plastic as its feature extractor is also trained with new data. The student model benefits from this plasticity since only the head is constrained to preserve similar outputs. While the architecture used for KD is different, the total loss of Dynamic Y-KD is similar as Faster-ILOD, except for the feature maps distillation since these are shared between the teacher and student networks. The loss for Dynamic Y-KD is as follows:

$$\mathcal{L}_{\text{DynYKD}} = \mathcal{L}_{faster} + \lambda_1 \mathcal{L}_{dist}^{box} + \lambda_3 \mathcal{L}_{dist}^{RPN}. \tag{9}$$

While more plasticity induces more forgetting, the ability to extract discriminative features of previous classes is preserved by reusing the feature extractors of previous steps. At deployment, Dynamic Y-KD uses all backbones that have been trained in previous steps and a merging mechanism combines the predictions made from the feature maps of each backbones. Interestingly, the results in [38] showed empirically that KD allows the head to still be compatible with previous features, although it has been exclusively trained for thousands of iterations on new classes. The main drawback of this method is that the model grows in parameters and computational costs for each new set of classes learned incrementally.

4. Experimental Setup

4.1. Evaluation Metrics

We compare the implemented methods using the mean average precision (mAP) at an intersection over union (IoU) threshold of 0.5, denoted mAP@0.5, or averaged over 10 thresholds spanning from 0.5 to 0.95 – termed as mAP@(0.5, 0.95). All results are measured after the final task of class-incremental learning. Following previous evaluation conventions in class-incremental learning [5, 38], we evaluate:

- Stability with mAP averaged over *base* classes (e.g. classes *1-42* in the *42-5* scenario of OPPD);

- Plasticity with mAP averaged over *new* classes (e.g. classes *43-47* in the *42-5* scenario of OPPD);

- Global performance with mAP averaged over *all* classes;

- Stability-Plasticity balance with mAP averaged over *intermediate* classes in multi-steps incremental learning scenarios (e.g. classes *2-6* in a *1-1* scenario that has a total of 7 classes).

Finally, we also introduce the mAP ratio with a joint training approach to discuss the degree to which class-incremental methods differ from the non-incremental upper-bound, namely joint training.

4.2. Implementation details

The experiments are conducted with the framework developed in [38]. We use Faster R-CNN with a ResNet-50 backbone initialized from ImageNet for all approaches. In addition to ILOD, Faster-ILOD and Dynamic Y-KD, we compare a naive fine-tuning approach as a baseline that does not use any mechanism to mitigate forgetting, and joint training which learns all classes simultaneously. The latter approach can be seen as an upper-bound that helps to evaluate the ability of continual learning models to tackle class-incremental challenges. In the following we describe the hyperparameters used by all methods for each dataset.

SDDD. For this dataset, we train each model for 10,000 iterations with SGD with a learning rate of 0.005 on the first set of classes. The learning rate is decayed by a factor of 0.1 at steps 8,000 and 9,500. For the next steps, the models are trained with a learning rate of 0.0001 for 2,500 iterations per class added. The batch size is set to 4 in all setups. We use the same split established in [2], which divides the 2,500 images into sets: 1,450 for training, 307 for validation, and 743 for testing. However, due to the small number of images per class, we merge the training and validation splits together to form a larger training set.

Method	6-1			5-2			4-3			3-4		
	1-6	7	1-7	1-5	6-7	1-7	1-4	5-7	1-7	1-3	4-7	1-7
Fine-tuning	17.4	35.8	20.0	10.2	35.2	17.3	6.9	33.3	18.2	12.7	20.3	17.0
ILOD [47]	39.7	29.6	38.3	20.0	27.5	22.2	22.4	22.2	22.3	22.9	15.9	18.9
Faster ILOD [40]	39.9	28.9	38.4	20.7	26.6	22.4	21.7	22.3	21.9	**23.9**	16.0	19.4
Dynamic Y-KD [38]	**40.1**	**31.8**	**38.9**	**22.0**	**32.8**	**25.1**	**25.2**	**26.9**	**25.9**	21.5	**20.4**	**20.9**
Joint Training	45.4	60.4	47.5	46.8	49.3	47.5	47.0	48.2	47.5	47.5	47.5	47.5

Table 1: mAP@(0.5, 0.95)% results of single-step incremental instance segmentation on the Strawberry Disease Detection Dataset.

Method	3-2				3-1				1-1			
	1-3	4-5	6-7	1-7	1-3	4-6	7	1-7	1	2-6	7	1-7
Fine-tuning	0.0	0.0	31.2	8.9	0.0	0.0	15.5	2.2	0.0	0.0	19.4	2.8
ILOD [47]	**23.5**	13.9	13.7	17.9	26.8	8.9	7.0	16.3	**13.9**	**8.0**	10.0	9.1
Faster ILOD [40]	22.8	13.4	14.3	17.7	25.3	8.9	6.9	15.7	16.5	2.7	6.4	5.2
Dynamic Y-KD [38]	23.3	**17.3**	**21.3**	**21.0**	**27.0**	**12.2**	**19.6**	**19.6**	13.0	7.1	**14.9**	**9.1**
Joint Training	47.5	45.7	49.3	47.5	47.5	43.3	60.4	47.5	44.7	45.5	60.4	47.5

Table 2: mAP@(0.5, 0.95)% results of multi-step incremental instance segmentation on the Strawberry Disease Detection Dataset.

OPPD. We train each model for 50,000 iterations with SGD with a learning rate of 0.005 on the first set of classes. For the next steps, the models are incremented for 10,000 iterations with a learning rate of 0.001. The batch size is also set to 4 in all setups. We divide the train and test splits by sampling a disjoint set of grow boxes for each species with their corresponding images. Similar to the previous dataset, we adapt the OPPD to incremental scenarios by sorting the plant species in alphabetical order and defining $b - n$ scenarios.

5. Results

5.1. Strawberry Disease Detection Dataset

Single-step incremental learning. We show in Table 1 the results for single-step incremental learning on the SDDD. We can see that catastrophic forgetting is particularly severe in *5-2*, *4-3* and *3-4* as the mAP of all approaches on old classes is less than half of the one obtained by joint training. Similarly, learning new classes incrementally is far from achieving the performance of joint training.

Nonetheless, we can see that Dynamic Y-KD outperforms ILOD and Faster ILOD in most settings both on previous and new classes. For instance, in *6-1*, Dynamic Y-KD obtains similar mAP on classes *1-6*, with 40.1% compared to 39.7% and 39.9% obtained by ILOD and Faster ILOD, respectively. But on the seventh category (*Powdery mildew leaf*), Dynamic Y-KD outperforms ILOD and Faster ILOD by $+2.2\%$ and $+2.9\%$, respectively. In *5-2*, *3-4* and *3-4*, the gap of mAP on new classes is even greater. These results confirm that the self-distillation employed by Dynamic Y-KD facilitates learning of new classes, while the dynamic architecture generally improves performance on

old classes. On all classes (*1-7*), Dynamic Y-KD outperforms other methods in every scenario.

Multi-steps incremental learning. Multi-steps incremental learning represents a more ambitious goal of class-incremental learning as it involves the ability to learn and remember several tasks successively. The results for various multi-steps incremental learning scenarios of the SDDD are shown in in Table 2.

We can see that forgetting is exacerbated and learning new classes is also complicated in these setups. In the *3-2* and *3-1* scenarios, all three continual learning methods perform similarly on the first set of classes. However, while ILOD and Faster ILOD also perform similarly on new classes, Dynamic Y-KD significantly outperforms them. For instance, in *3-2*, Dynamic Y-KD obtains $+3.4\%$ and $+7.0\%$ mAP on both sets of novel classes compared to the second best approaches. In *3-1*, Dynamic Y-KD outperforms ILOD by $+12.6\%$ mAP on the last class while suffering less from forgetting.

However, the results are very far from joint training, especially in the *1-1* scenario where the best performances only reach $\frac{9.1\%}{47.5\%} = 19.2\%$ of the global performance of joint training. This highlights the difficulty of class-incremental learning, especially on difficult tasks involving agricultural imagery. While current continual learning methods can have reasonable performances in single-step incremental learning, there is a lot of room for improvement on long-term incremental learning.

5.1.1 Open Plant Phenotyping Database

We now report the mAP@0.5% of single-step incremental learning scenarios on the OPPD in Table 3. We can ob-

	42-5			37-10			27-20		
Method	**1-42**	**43-47**	**1-47**	**1-37**	**38-47**	**1-47**	**1-27**	**28-47**	**1-47**
Fine-tuning	5.6	22.8	7.5	2.5	22.4	6.8	3.6	19.8	10.6
ILOD [47]	19.5	20.9	19.6	17.3	20.0	17.9	16.7	15.1	16.0
Faster ILOD [40]	19.5	20.8	19.6	**17.5**	19.8	18.0	16.7	15.1	16.0
Dynamic Y-KD [38]	**21.6**	**22.5**	**21.7**	**17.5**	**21.8**	**18.4**	**17.5**	**17.1**	**17.3**
Joint Training	23.9	29.0	24.5	23.6	27.7	24.5	24.3	24.8	24.5

Table 3: mAP@0.5% results of single-step incremental instance segmentation on the Open Plant Phenotyping Database.

serve from the results that this dataset is very challenging. Indeed, the OPPD contains many classes, large amounts of instances on each image, and the plants have large intra-class and small inter-class variability (see Fig. 3). Similar to the results on SDDD, ILOD and Faster-ILOD perform comparatively in all three setups, while Dynamic Y-KD obtains better results. Interestingly, Dynamic Y-KD reaches similar mAP@0.5% than fine-tuning, while still suffering less from forgetting than other methods. Indeed, Dynamic Y-KD obtains 22.5%, 21.8% and 17.1% compared to 22.8%, 22.4% and 19.8% by fine-tuning on new classes in *42-5*, *37-10* and *27-20* scenarios, respectively.

Overall, Dynamic Y-KD reaches $\frac{21.7}{24.5} = 88.6\%$ of the joint training performance on all classes in *42-5*. When more classes are involved, Dynamic Y-KD only reaches $\frac{18.4}{24.5} = 75.1\%$ and $\frac{17.3}{24.5} = 70.6\%$ of the performance obtained by joint training in *37-10* and *27-20*, respectively.

5.2. Discussion

Together, the results of class-incremental learning methods compared to joint training highlight the difficulty of such scenario on agricultural imagery. For instance, on the Strawberry Disease Detection Dataset in the *5-2* scenario, the best-performing method obtains 22.0% on classes *1-5* whereas joint training obtains 46.8% (see Tab. 1), indicating a strong effect of forgetting.

Future work should investigate the underlying reasons. We hypothesize that the small inter-class variability is especially challenging for class-incremental learning. Notably, as classes 4 (Gray mold) and 6 (Powdery mildew) are relatively similar (see Fig. 2) in this dataset, features that are learned to detect the 6th class might strongly interfere with those that were used to detect the 4th class learned in a previous step. Consequently, the incremental model might learn to suppress previous knowledge to reduce interference and thereby optimize the current task.

Future work should also investigate a combination of rehearsal techniques and other continual learning strategies. For instance, the hybrid approach of Dynamic Y-KD combined with a small memory bank of previous examples might allow preserving more discriminative features of previous classes and reduce forgetting.

6. Conclusion

This work is among the first to tackle continual object detection for tasks related to agriculture. To this aim, we adapted two public datasets to simulate incremental learning scenarios in which classes are learned in two or more distinct steps. In each step, the models only have access to current classes for training.

To address the problem of catastrophic forgetting faced by deep learning models, we compared ILOD, Faster-ILOD and Dynamic Y-KD. All three approaches leverage different forms of knowledge distillation to reduce knowledge loss during incremental learning. Dynamic Y-KD reduces the constraining factor of knowledge distillation by sharing the same backbone between the teacher and student networks, which proved to be more effective at learning new classes on both datasets. Moreover, as Dynamic Y-KD uses a parameter isolation strategy in which previous and new backbones are used at inference, catastrophic forgetting is also mitigated at a cost of increased computation.

The experiments conducted in this work showed that the three continual learning methods effectively addressed the issue of forgetting while learning new classes, yet their performances are far from the joint training upper-bound. This highlights the potential for further improvements in the field of continual object detection, particularly in the context of agricultural imagery.

We hypothesized that the inherent challenges posed by the small intra-class and large inter-class variability characteristic of plant images exacerbate the problem of catastrophic forgetting. Future work should explore rehearsal approaches to reduce forgetting, combined with hybrid strategies such as Dynamic Y-KD to improve learning of new classes.

Acknowledgment

References

[1] Mahmoud Abdulsalam and Nabil Aouf. Deep weed detector/classifier network for precision agriculture. In *2020*

28th Mediterranean Conference on Control and Automation (MED), pages 1087–1092. IEEE, 2020. 3

[2] Usman Afzaal, Bhuwan Bhattarai, Yagya Raj Pandeya, and Joonwhoan Lee. An instance segmentation model for strawberry diseases based on mask r-cnn. *Sensors*, 21(19):6565, 2021. 2, 4, 6

[3] Bhuwan Bhattarai, Yagya Raj Pandeya, and Joonwhoan Lee. An incremental learning for plant disease classification. In *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1147–1149. IEEE, 2021. 1, 3

[4] Pieter M Blok, Gert Kootstra, Hakim Elchaoui Elghor, Boubacar Diallo, Frits K van Evert, and Eldert J van Henten. Active learning with maskal reduces annotation effort for training mask r-cnn on a broccoli dataset with visually similar classes. *Computers and Electronics in Agriculture*, 197:106917, 2022. 2, 3

[5] Fabio Cermelli, Antonino Geraci, Dario Fontanel, and Barbara Caputo. Modeling missing annotations for incremental learning in object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3700–3710, 2022. 4, 6

[6] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9233–9242, 2020. 3

[7] Songhee Cho, Taehyeong Kim, Dae-Hyun Jung, Soo Hyun Park, Yunseong Na, Yong Seok Ihn, and KangGeon Kim. Plant growth information measurement based on object detection and image fusion using a smart farm robot. *Computers and Electronics in Agriculture*, 207:107703, 2023. 3

[8] Solemane Couliably, Bernard Kamsu-Foguem, Dantouma Kamissoko, and Daouda Traore. Deep learning for precision agriculture: A bibliometric analysis. *Intelligent Systems with Applications*, page 200102, 2022. 1

[9] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021. 1

[10] United Nations Population Division. The 2022 revision of world population prospects. United Nations Department of Economic and Social Affairs, New York City, 2022. https://population.un.org/wpp/, (accessed Mar. 14, 2023). 1

[11] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Plop: Learning without forgetting for continual semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4040–4050, 2021. 3

[12] Lei Du, Yaqin Sun, Shuo Chen, Jiedong Feng, Yindi Zhao, Zhigang Yan, Xuewei Zhang, and Yuchen Bian. A novel object detection model based on faster r-cnn for spodoptera frugiperda according to feeding trace of corn leaves. *Agriculture*, 12(2):248, 2022. 3

[13] 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*, 88:303–308, 2009. 1

[14] Shirin Ghatrehsamani, Gaurav Jha, Writuparna Dutta, Faezeh Molaei, Farshina Nazrul, Mathieu Fortin, Sangeeta Bansal, Udit Debangshi, and Jasmine Neupane. Artificial intelligence tools and techniques to combat herbicide resistant weeds—a review. *Sustainability*, 15(3):1843, 2023. 3

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2, 4

[16] Gaurav Jha, Floyid Nicolas, Radomir Schmidt, Kosana Suvočarev, Dawson Diaz, Isaya Kisekka, Kate Scow, and Mallika A Nocco. Irrigation decision support systems (idss) for california's water–nutrient–energy nexus. *Agronomy*, 12(8):1962, 2022. 1

[17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 3

[18] Anand Koirala, Kerry B Walsh, Zhenglin Wang, and Cheryl McCarthy. Deep learning–method overview and review of use for fruit detection and yield estimation. *Computers and electronics in agriculture*, 162:219–234, 2019. 3

[19] G Hari Krishnan and T Rajasenbagam. A comprehensive survey for weed classification and detection in agriculture lands. *Journal of Information Technology*, 3(4):281–289, 2021. 3

[20] Jaesu Lee, Haseeb Nazki, Jeonghyun Baek, Youngsin Hong, and Meonghun Lee. Artificial intelligence approach for tomato detection and mass estimation in precision agriculture. *Sustainability*, 12(21):9138, 2020. 3

[21] Simon Leminen Madsen, Solvejg Kopp Mathiassen, Mads Dyrmann, Morten Stigaard Laursen, Laura-Carlota Paz, and Rasmus Nyholm Jørgensen. Open plant phenotype database of common weeds in denmark. *Remote Sensing*, 12(8):1246, 2020. 2, 4, 5

[22] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information fusion*, 58:52–68, 2020. 1

[23] Wei Li, Qingbo Wu, Linfeng Xu, and Chao Shang. Incremental learning of single-stage detectors with mining memory neurons. In *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, pages 1981–1985. IEEE, 2018. 3

[24] Yang Li and Xuewei Chao. Ann-based continual classification in agriculture. *Agriculture*, 10(5):178, 2020. 1, 3

[25] Zhenbo Li, Ruohao Guo, Meng Li, Yaru Chen, and Guangyao Li. A review of computer vision technologies for plant phenotyping. *Computers and Electronics in Agriculture*, 176:105672, 2020. 1

[26] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 3

[27] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 2021. 1

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

[29] Martina Lippi, Niccolò Bonucci, Renzo Fabrizio Carpio, Mario Contarini, Stefano Speranza, and Andrea Gasparri. A yolo-based pest detection system for precision agriculture. In *2021 29th Mediterranean Conference on Control and Automation (MED)*, pages 342–347. IEEE, 2021. 3

[30] Jun Liu and Xuewei Wang. Plant diseases and pests detection based on deep learning: a review. *Plant Methods*, 17:1–18, 2021. 1

[31] Shenglian Lu, Zhen Song, Wenkang Chen, Tingting Qian, Yingyu Zhang, Ming Chen, and Guo Li. Counting dense leaves under natural environments via an improved deep-learning-based object detection algorithm. *Agriculture*, 11(10):1003, 2021. 3

[32] 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. 1, 4

[33] Andrea Maracani, Umberto Michieli, Marco Toldo, and Pietro Zanuttigh. Recall: Replay-based continual learning in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7026–7035, 2021. 3

[34] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 1

[35] Angelo G Menezes, Gustavo de Moura, Cézanne Alves, and André CPLF de Carvalho. Continual object detection: a review of definitions, strategies, and challenges. *Neural Networks*, 2023. 1, 3

[36] Thomas J Monaco, Stephen C Weller, and Floyd M Ashton. *Weed science: principles and practices*. John Wiley & Sons, 2002. 1, 3

[37] Salima Ouadfel, Wafa Mousser, Ismail Ghoul, and Abdelmalik Taleb-Ahmed. Incremental deep learning model for plant leaf diseases detection. In *Artificial Neural Networks for Renewable Energy Systems and Real-World Applications*, pages 207–222. Elsevier, 2022. 1, 3

[38] Mathieu Pagé-Fortin and Brahim Chaib-draa. Dynamic y-kd: A hybrid approach to continual instance segmentation. *arXiv preprint arXiv:2303.06015*, 2023. 2, 3, 4, 6, 7, 8

[39] Diego Inácio Patrício and Rafael Rieder. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Computers and electronics in agriculture*, 153:69–81, 2018. 1

[40] Can Peng, Kun Zhao, and Brian C Lovell. Faster ilod: Incremental learning for object detectors based on faster rcnn. *Pattern recognition letters*, 140:109–115, 2020. 2, 3, 4, 5, 7, 8

[41] Ildar Rakhmatulin, Andreas Kamilaris, and Christian Andreasen. Deep neural networks to detect weeds from crops in agricultural environments in real-time: a review. *Remote Sensing*, 13(21):4486, 2021. 3

[42] 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. 4, 5

[43] Adrian Salazar-Gomez, Madeleine Darbyshire, Junfeng Gao, Elizabeth I Sklar, and Simon Parsons. Beyond map: Towards practical object detection for weed spraying in precision agriculture. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9232–9238. IEEE, 2022. 3

[44] Farzin Shabani, Mohsen Ahmadi, Lalit Kumar, Samaneh Solhjouy-fard, Mahyat Shafapour Tehrany, Fariborz Shabani, Bahareh Kalantar, and Atefeh Esmaeili. Invasive weed species' threats to global biodiversity: Future scenarios of changes in the number of invasive species in a changing climate. *Ecological Indicators*, 116:106436, 2020. 1, 3

[45] Tawseef Ayoub Shaikh, Tabasum Rasool, and Faisal Rasheed Lone. Towards leveraging the role of machine learning and artificial intelligence in precision agriculture and smart farming. *Computers and Electronics in Agriculture*, 198:107119, 2022. 1

[46] Jeng-Lun Shieh, Qazi Mazhar ul Haq, Muhamad Amirul Haq, Said Karam, Peter Chondro, De-Qin Gao, and Shanq-Jang Ruan. Continual learning strategy in one-stage object detection framework based on experience replay for autonomous driving vehicle. *Sensors*, 20(23):6777, 2020. 3

[47] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE international conference on computer vision*, pages 3400–3409, 2017. 2, 3, 4, 5, 7, 8

[48] ChauChung Song, ChihLun Wang, and YiFeng Yang. Automatic detection and image recognition of precision agriculture for citrus diseases. In *2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, pages 187–190. IEEE, 2020. 3

[49] Marco Sozzi, Silvia Cantalamessa, Alessia Cogato, Ahmed Kayad, and Francesco Marinello. Grape yield spatial variability assessment using yolov4 object detection algorithm. In *Precision agriculture'21*, pages 193–198. Wageningen Academic Publishers, 2021. 3

[50] Thomas Van Klompenburg, Ayalew Kassahun, and Cagatay Catal. Crop yield prediction using machine learning: A systematic literature review. *Computers and Electronics in Agriculture*, 177:105709, 2020. 1

[51] Eli Verwimp, Matthias De Lange, and Tinne Tuytelaars. Rehearsal revealed: The limits and merits of revisiting samples in continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9385–9394, 2021. 3

[52] Ranjeet Walia, Shivam Sharma, and Swapnil Shrivastava. A methodology for early detection of plant diseases using real time object detection algorithm. In *Smart and Sustainable*

Agriculture: First International Conference, SSA 2021, Virtual Event, June 21-22, 2021, Proceedings 1, pages 122–139. Springer, 2021. 3

[53] Rujing Wang, Liu Liu, Chengjun Xie, Po Yang, Rui Li, and Man Zhou. Agripest: A large-scale domain-specific benchmark dataset for practical agricultural pest detection in the wild. *Sensors*, 21(5):1601, 2021. 3

[54] Omer Wosner, Guy Farjon, and Aharon Bar-Hillel. Object detection in agricultural contexts: A multiple resolution benchmark and comparison to human. *Computers and Electronics in Agriculture*, 189:106404, 2021. 3

[55] Zhangnan Wu, Yajun Chen, Bo Zhao, Xiaobing Kang, and Yuanyuan Ding. Review of weed detection methods based on computer vision. *Sensors*, 21(11):3647, 2021. 1

[56] Jing Zhang, Manoj Karkee, Qin Zhang, Xin Zhang, Majeed Yaqoob, Longsheng Fu, and Shumao Wang. Multi-class object detection using faster r-cnn and estimation of shaking locations for automated shake-and-catch apple harvesting. *Computers and Electronics in Agriculture*, 173:105384, 2020. 3

[57] Yang Zhang, Chenglong Song, and Dongwen Zhang. Deep learning-based object detection improvement for tomato disease. *IEEE access*, 8:56607–56614, 2020. 3

[58] Wei Zhao, William Yamada, Tianxin Li, Matthew Digman, and Troy Runge. Augmenting crop detection for precision agriculture with deep visual transfer learning—a case study of bale detection. *Remote Sensing*, 13(1):23, 2020. 3

[59] Yang-Yang Zheng, Jian-Lei Kong, Xue-Bo Jin, Ting-Li Su, Ming-Jun Nie, and Yu-Ting Bai. Real-time vegetables recognition system based on deep learning network for agricultural robots. In *2018 Chinese Automation Congress (CAC)*, pages 2223–2228. IEEE, 2018. 3