Modeling Missing Annotations for Incremental Learning in Object Detection

Modeling Missing Annotations for Incremental Learning in Object Detection

Fabio Cermelli[1,2], Antonino Geraci[1], Dario Fontanel[1], Barbara Caputo[1]
[1]Politecnico di Torino, [2]Italian Institute of Technology
fabio.cermelli@polito.it

Abstract

Despite the recent advances in the field of object detection, common architectures are still ill-suited to incrementally detect new categories over time. They are vulnerable to catastrophic forgetting: they forget what has been already learned while updating their parameters in absence of the original training data. Previous works extended standard classification methods in the object detection task, mainly adopting the knowledge distillation framework. However, we argue that object detection introduces an additional problem, which has been overlooked. While objects belonging to new classes are learned thanks to their annotations, if no supervision is provided for other objects that may still be present in the input, the model learns to associate them to background regions. We propose to handle these missing annotations by revisiting the standard knowledge distillation framework. Our approach outperforms current state-of-the-art methods in every setting of the Pascal-VOC dataset. We further propose an extension to instance segmentation, outperforming the other baselines.

1. Introduction

Object detection is a key task in computer vision that has seen significant development in recent years. The advances were made possible by the rise of deep neural network architectures [3, 19, 24, 33, 47, 48], which improved results while reducing computation time. Despite the advances, these architectures assume that they already know all of the classes they will encounter and are not designed to incrementally update their knowledge to learn new classes over time. A naïve solution would be to restart the training process from the beginning, gathering a new dataset with all of the classes and retraining the architecture. However, this is impractical because it would necessarily require a significant computational overhead to re-learn the previously learned classes, as well as the use of previous training data that may no longer be available, for example due to privacy concerns or intellectual property rights.

A better solution is to use incremental learning and update the models directly to extend their knowledge to new classes by training only on new data and avoiding catastrophic forgetting [39]. Incremental learning has primarily been studied in the context of image classification [2, 14, 18, 28, 31, 46, 49] but it has only recently been applied to more complex tasks like object detection [10, 22, 43, 51, 59] and semantic segmentation [5–8, 13, 40, 41]. Performing incremental learning in object detection (ILOD) poses additional challenges because each image contains multiple objects and, following the definition in [51], only objects belonging to new classes are annotated while the rest (objects belonging to either old or future classes) are ignored, introducing missing annotations (see Fig. 1).

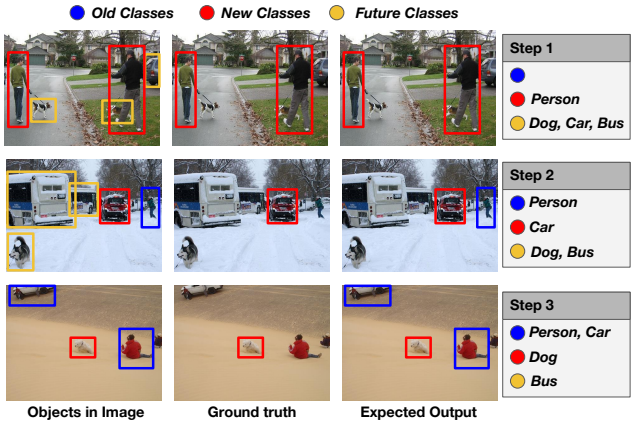

Figure 1. An illustration of the missing annotation issue of object detection in different time steps. At training step t, the annotations are provided only for new classes (red), while all the other objects, both from old (blue) and future (yellow) steps are not annotated.

Previous research has concentrated on introducing regularizations to prevent catastrophic forgetting, but the impact of missing annotations has been overlooked. Regions without annotations, in particular, are commonly considered as background areas, and the model assigns them to a special *background* class. As a result, objects that are not annotated will be associated with the background, exacerbating catastrophic forgetting in old classes and making training more difficult in future classes.

To overcome this issue, inspired by [6], we revisit the common knowledge distillation framework in ILOD [43,51, 59] proposing MMA, that **M**odels the **M**issing **A**nnotations in both the classification and distillation losses. We flexibly allow the model to predict either an old class or the background on any region not associated with an annotation on the classification loss to alleviate catastrophic forgetting. Alternatively, because current classes may have been annotated as background in a previous learning step, we revisit the distillation loss, matching the teacher model's background probability with the probability of having either a new class or the background, allowing new classes to be learned more easily. On the Pascal-VOC dataset [16], we demonstrate the utility of our method by examining a variety of single-step and multi-step tasks. Without using any image from previous training steps, we show that our method outperforms the current state-of-the-art.

Finally, we show that by adding an additional knowledge distillation term to our framework, we can easily extend it to the task of instance segmentation. On the Pascal SBD 2012 dataset [17], we show that our method outperforms the other baselines.

To summarize, the contributions of this paper are as follows:

- We identify the peculiar missing annotations issue in incremental learning for object detection.
- We propose to revisit the standard knowledge distillation framework to cope with the missing annotations, showing that our proposed MMA outperforms previous methods on multiple incremental settings.
- We extend our method to instance segmentation and we show that it outperforms all the other baselines. [1]

2. Related work

Object Detection. Object detection architectures can be mainly distinguished in two categories: one-stage detectors [3,35,47,52,53,61] and two-stage detectors [19,20,24, 32,48]. Two-stage detectors are usually superior in performance but are less efficient, implementing two subsequent steps to perform detection: the model first extract regions of interest (RoIs) employing either a neural network [48] or an external region proposer [19] and then use a MLP on the RoIs to obtain the final classification and bounding box regression. Differently, one-stage detectors directly predict the final output, without requiring to predict RoIs. These architectures are undoubtedly powerful in a standard, offline setting but they are not suited to incrementally add new classes over time. In this work, we focus on extending two-stage methods, in particular the Faster R-CNN [48], to extend its knowledge on new categories without forgetting the previous knowledge in absence of the original data.

[1] Code can be found here https://github.com/fcdl94/MMA.

Incremental Learning. The problem of catastrophic forgetting [39] has been extensively studied in the image classification task and recently extended to semantic segmentation. Previous works can be divided in three categories: rehearsal-based [4, 26, 42, 46, 50, 54], regularization-based [9, 12, 28, 31, 58] and parameter isolation-based [37, 38, 49]. Rehearsal-based methods either store [4,26,46,55] or generate [42,50,54] examples of previous tasks, which are used to compensate for the lack of previous data during the training phase of the new task. Parameter isolation-based methods assign a subset of the parameters to each task and prevent them to change to avoid forgetting. Regularization-based methods can be divided in prior-focused and data-focused. The former [2, 9, 28, 58] relies on knowledge stored in parameters value, constraining the learning of new tasks by penalizing changes of important old parameters. The latter [6, 12, 14, 15, 18, 26, 31, 56] exploits distillation [25] and uses the distance between the activation produced by the old network and the new one as a regularization term to prevent catastrophic forgetting. In this work, we focus on the data-focused regularization-based knowledge distillation approach by adapting it in the object detection context while modeling the missing annotations issue. We note that [6] identified a problem similar to the missing annotations in incremental semantic segmentation called background shift. We take inspiration from it to address the missing annotation problem in object detection.

Incremental Learning in Object Detection. Incremental learning in Object detection has witnessed more attention in last years. A pioneer work in this task is [51], that proposes a framework based on two-stage object detectors by performing knowledge distillation on the output of Fast R-CNN [19]. Inspired by this work, some methods extend the distillation framework on the Faster R-CNN [48] architecture by adding distillation terms on the intermediate feature maps [10, 36, 43, 57, 59] and proposing to further avoid forgetting on region proposal network [10, 22, 43, 59]. Interestingly [59] proposed a pseudo-positive-aware sampling algorithm to identify regions belonging to old classes and preventing them to be sampled as background regions. However, it only provides a partial solution for the missing annotation since it does not consider them in the distillation term nor the confidence of the model. Other methods [1, 21, 27, 29] focused on rehearsal methods to maintain the old task knowledge, either performing replay of the intermediate features [1] or the images [21, 21, 30]. Differently, [34] proposes a parameter isolated method extending EWC [28] in the context of object detection. Finally, a few works explored incremental learning utilizing one-stage architectures [30, 44, 45]. In this work, we focus on proposing a distillation framework for two-stage architectures by explicitly modeling the missing annotations about object not belonging to the current training step.

3. Method

3.1. Problem Definition and Preliminaries

The goal of object detection is to train a model able to detect objects, *i.e*. localize and classify them by producing a rectangular box and a class label. In this work, we focus on detection model in the R-CNN [19, 24, 48] family. A detection model, denoted $\mathcal{F}_\theta$ with parameters θ, is composed by three components: a feature extractor $\mathcal{F}_{FE}$, a region proposal network (RPN) $\mathcal{F}^{RPN}_\theta$, and a classification head $\mathcal{F}^{RCN}_\theta$. Denoting with x an image, the feature extractor produces a dense feature map. The map is forwarded to the RPN with the goal of producing a set of N rectangular regions of interest (RoIs), each associated with a binary objectness score. The N RoIs are then applied to the feature map and classified by the classification head that produces, for each RoI, the class probabilities $p \in \mathbb{R}^{|\mathcal{C}|+1}$, indicating with $\mathcal{C}$ the set of classes, and the rectangular boxes $r \in \mathbb{R}^{4|\mathcal{C}|}$, one for each class. We note that the classifier also outputs a class score for the background to indicate that no objects are present in the RoI.

In incremental learning for object detection (ILOD) the training is performed over multiple *learning steps*, each one introducing a new set of classes to be detected. Formally, in the t-th training step, a detection model $\mathcal{F}_{\theta^t}$ is updated to learn a set of classes $\mathcal{Y}^t$ employing a training set $\mathcal{D}^t$. We note that while an image in the training set $\mathcal{D}^t$ can contain multiple objects of different classes, following the ILOD protocol [51] *only* annotations for classes in $\mathcal{Y}^t$ are provided. Moreover, at training step t the old training sets are not available. After the t-th step, the model $\mathcal{F}_{\theta^t}$ is expected to produce prediction for all the classes seen so far, *i.e*. its output should consider the classes in $\mathcal{C}^t = \cup_{t'=1}^{t} \mathcal{Y}^{t'}$. We note that $\mathcal{Y}^i \cap \mathcal{Y}^j = \emptyset$ for any $i, j \leq t$ and $i \neq j$.

Faster R-CNN. In the standard Faster R-CNN [48] training is performed minimizing a multi-task loss as follows:

$$\ell_{faster} = \ell^{RPN}_{cls} + \ell^{RPN}_{reg} + \ell^{RCN}_{cls} + \ell^{RCN}_{reg}. \quad (1)$$

The first two terms are the classification and regression loss on the RPN [48], while the latter are applied on the classification head output [19]. Please refer to [19, 48] for additional details on the training of Faster R-CNN.

3.2. MMA: Modeling the Missing Annotations

Despite its strength, Faster R-CNN is not well suited to updating its weights in order to learn new classes. Fine-tuning the model on $\mathcal{D}^t$ using Eq. (1), in particular, causes the model to forget everything it has learned, resulting in catastrophic forgetting [39]. To address this, previous research [10, 22, 43, 51, 59] proposed the use of knowledge distillation [25, 31], in which, at the training step, the *student* model $\mathcal{F}_{\theta^t}$ is forced to mimic the output of the *teacher* model $\mathcal{F}_{\theta^{t-1}}$, *i.e*. the model at the previous training step.

Previous research, while addressing forgetting, did not address the issue of missing annotations. At time step t the dataset $\mathcal{D}^t$ provides annotations only for objects in $\mathcal{Y}^t$ and other objects present in the image, belonging either to past or future classes, are not annotated. Following the standard detection pipeline, any RoI that does not match a ground truth annotation is associated to the background. This introduces two issues: (i) if the RoI contains an object of an old class, the model learns to predict it as background, exacerbating the forgetting; (ii) when the RoI contains an object that will be learned in the future, the model learns to consider it as background, making harder to learn new classes when presented. The missing annotation issue is similar to the background shift presented in [6] in the context of incremental learning for semantic segmentation. In the following, we show how to adapt the equations proposed by [6] in incremental learning for object detection.

Unbiased Classification Loss. The classification loss ℓ^{RCN}_{cls} in the Faster R-CNN has the goal to force the network to produce the correct class label for the RoIs. In detail, given a sampled set of N RoIs generated by the RPN and matched with a ground truth label (positive RoI) or with the background (negative RoI), the loss is computed as:

$$\ell^{RCN}_{cls} = \frac{1}{N}\sum_{i=1}^{N} z_i \Big(\sum_{c\in\mathcal{C}^t} \bar{y}_i^c \log(p_i^c)\Big) + (1 - z_i)\log(p_i^b), \quad (2)$$

where z_i is 1 for a positive RoI and 0 otherwise, $\bar{y}_i$ is the one-hot class label (1 for the ground truth class, 0 otherwise), and p_i^b indicates the probability for the background class for the i-th RoI.

The Eq. (2) does not consider that only information about novel classes is available in the ground truth because it was designed for standard object detection. The problem is that all other objects in the image that are not associated with any ground-truth are treated as a negative RoI and the model learns to predict the background class on them. This problem is especially harmful for objects of old classes because it causes the model to forget the object's correct class and replace it with the background class, resulting in severe catastrophic forgetting.

To avoid this issue, we modify Eq. (2) as follows:

$$\begin{aligned}\ell^{RCN}_{cls} = \frac{1}{N}\sum_{i=1}^{N} z_i \Big(\sum_{c\in\mathcal{Y}^t} \bar{y}_i^c \log(p_i^c)\Big) + \\ (1 - z_i)\log(p_i^b + \sum_{o\in\mathcal{C}^{t-1}} p_i^o),\end{aligned} \quad (3)$$

where p_i^c is the probability of class c for query i, $\mathcal{Y}^t$ are the new classes at t and $\mathcal{C}^{t-1}$ are all the classes seen before t. Using Eq. (3) the model learns new classes on the positive RoIs ($z_i = 1$) while preventing the background to supersede

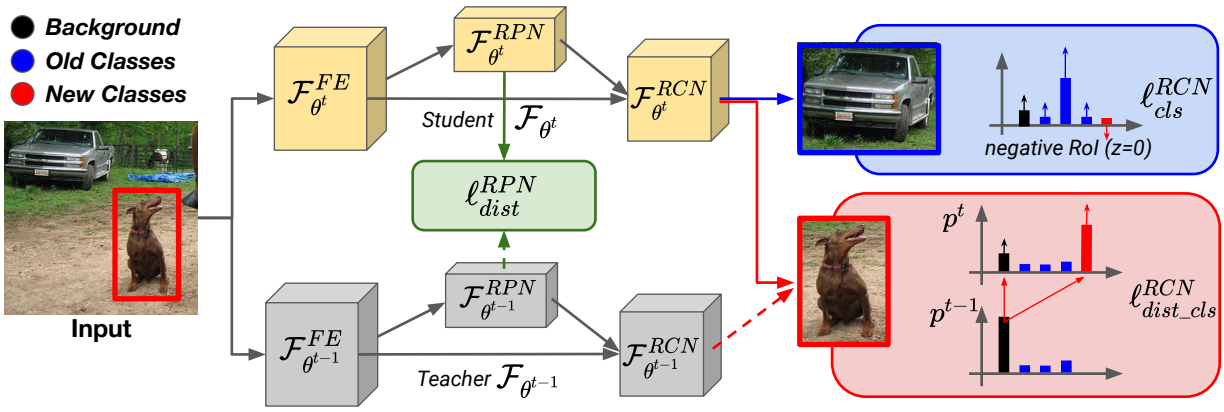

Figure 2. Overview of MMA, highlighting its contributions. Given an image, it is forwarded on the student (top) and teacher (bottom) models. The blue box illustrates the behavior of unbiased cross entropy loss on a negative RoI (*i.e.* RoI without annotation): the model maximizes the probability of having either the background or an old class. In the red box, we show the effect of the unbiased distillation loss on the classification output for a new class region: it associates the teacher background with either the student background or a new class. Lastly, in green, it is reported the RPN distillation loss.

the old classes: instead of forcing the background class on every negative RoI ($z_i = 0$), as in Eq. (2), it forces the model to predict either the background or any old class by maximizing the sum of their probabilities. An illustration is reported in the blue box of Fig. 2.

Unbiased Knowledge Distillation. A common solution to avoid forgetting is to add two knowledge distillation loss terms to the training objective [10, 22, 43, 59]:

$$\ell = \ell_{faster} + \lambda^1 \ell^{RCN}_{dist} + \lambda^2 \ell^{RPN}_{dist}, \tag{4}$$

where λ^1, λ^2 are hyper-parameters.

The goal of ℓ^{RCN}_{dist} is to maintaining the knowledge about old classes on the classification head. Previous works [43, 51] force the student model to output classification scores and box coordinates for old classes close to the teacher employing an L2 loss. However, they ignore the missing annotations, *i.e.* the new classes have been observed in previous steps but, since they had been observed without annotations, they have been associated to the background class. The teacher would predict an high background score for new classes RoIs, and forcing the student to mimic its behavior would make harder to learn new classes, contrasting the classification loss. Taking model the missing annotations, we formulate the distillation loss as:

$$\ell^{RCN}_{dist} = \frac{1}{N}\sum_{i=1}^{N} \ell^{RCN}_{dist\_cls}(i) + \ell_{smooth\_l1}(r^t_i, r^{t-1}_i), \tag{5}$$

$$\ell^{RCN}_{dist\_cls}(i) = \frac{1}{|\mathcal{C}^{t-1}|+1}\Big(p^{b,t-1}_i \log(p^{b,t}_i + \sum_{j\in\mathcal{Y}^t} p^{j,t}_i) + \sum_{c\in\mathcal{C}^{t-1}} p^{c,t-1}_i \log(p^{c,t}_i)\Big), \tag{6}$$

where $p^{k,t-1}_i$, r^{t-1}_i and $p^{k,t}_i$, r^t_i indicates, respectively, the classification and regression output for the proposal i and class k of the teacher and student model, and b is the background class. While the second term of Eq. (5) has been used in previous works [43, 51] and considers the box coordinates, we propose to modify the first term that is responsible to handle the classification scores. To model the missing annotations, Eq. (6) uses the all the class probabilities of the student model to match the teacher ones: the old classes $\mathcal{C}^{t-1}$ are kept unaltered among student and teacher models, while the background of the teacher $p^{b,t-1}_i$ is associated with either a new class or the background in the student. With Eq. (6), when the teacher predicts an high background probability for a RoI belonging to a new class, the student is not forced to mimic its behavior but it can consolidate its new knowledge and predict the correct class. An illustration is reported in the red box of Fig. 2.

On the other hand, ℓ^{RPN}_{dist} goal is to avoid forgetting on the RPN output. Since annotation for old classes are not available, the RPN learns to predict an high objectness score only on RoIs belonging to new classes. To force the RPN to maintain an high objectness score for regions belonging to old classes, we use the loss proposed by [43]. The student is forced to mimic the teacher only on regions belonging to old classes, *i.e.* where the teacher score is greater than the student one. Considering A regions, we compute ℓ^{RPN}_{dist} as:

$$\ell^{RPN}_{dist} = \frac{1}{A}\sum_{i=1}^{A} \mathbb{1}_{[s^t_i \geq s^{t-1}_i]} ||s^t_i - s^{t-1}_i|| + \mathbb{1}_{[s^t_i \geq s^{t-1}_i + \tau]} ||\omega^t_i - \omega^{t-1}_i||, \tag{7}$$

where s^t_i is the objectness score and ω^t_i the coordinates of $\mathcal{F}^{RPN}_{\theta^t}$ on the i-th proposal, $||\cdot||$ is the euclidean distance, τ is an hyperparameter, and $\mathbb{1}$ is the indicator function equal

to 1 when the condition on the brackets is verified and 0 otherwise. Note that when $s_i^t > s_i^{t-1}$, the teacher produces an objectness score greater then the student and the proposal is probably containing an old class. Differently, when $s_i^t \geq s_i^{t-1}$, the proposal is likely belonging to a new class and forcing the student to mimic the teacher score may introduce errors that hamper the performance on new classes.

3.3. Extension to Instance Segmentation

The goal of instance segmentation is to produce a precise pixel-wise mask for each object in the image. To produce masks we rely on Mask R-CNN [24], that extends the Faster R-CNN introducing a mask head $\mathcal{F}_\theta^{MASK}$. It produces, for each RoI, an additional binary segmentation mask with shape $|\mathcal{C}| \times h \times w$, where $\mathcal{C}$ is the set of classes and h,w is the mask resolution. To train the mask head, [24] introduces an additional loss term that is summed to the multi-task loss in Eq. (1). Formally, Mask R-CNN objective is:

$$\ell_{mask} = \ell_{faster} + \ell_{cls}^{MASK}, \tag{8}$$

where ℓ_{cls}^{MASK} is the per-pixel binary cross-entropy loss between the $\mathcal{F}_\theta^{MASK}$ output and the binary mask of the ground truth class. Please refer to [24] for details.

Despite the method presented in Sec. 3.2 already accounts for forgetting on the detection head, by applying Eq. (8) we incur the risk to forget how to segment past objects while learning the new ones. For this reason, we further extend Eq. (4) to add a knowledge distillation term on the mask head. Formally, in instance segmentation we employ the following training objective:

$$\ell = \ell_{mask} + \lambda_1 \ell_{dist}^{RCN} + \lambda_2 \ell_{dist}^{RPN} + \lambda_3 \ell_{dist}^{MASK}, \tag{9}$$

where λ_1, λ_2, λ_3 are hyper-parameters.

ℓ_{dist}^{MASK} has the goal of keeping the segmentation mask for old classes close to the output of the teacher model. In particular, we employ a per-pixel binary cross-entropy loss between the teacher model masks and the student ones. Formally, denoting as $m_{c,i}^t$ the segmentation mask produced by $\mathcal{F}_{\theta^t}^{MASK}$ for the class c at pixel i, we compute

$$\ell_{dist}^{MASK} = \frac{1}{|I||\mathcal{C}^{t-1}|} \sum_{i \in I} \sum_{c \in \mathcal{C}^{t-1}} m_{c,i}^{t-1} \log(m_{c,i}^t) + (1 - m_{c,i}^{t-1}) \log(1 - m_{c,i}^t), \tag{10}$$

where I is the set of pixels and $|I| = h \times w$. We note that Eq. (10) is computed only on the segmentation masks belonging to old classes in $\mathcal{C}^{t-1}$, while the masks belonging to the new ones are not considered.

4. Experiments

4.1. Experimental Protocol

We evaluate MMA on the Pascal-VOC dataset. In particular, following previous works, we employ PASCAL-VOC 2007 [16] for object detection. It is a widely used benchmark that includes 20 foreground object classes and consists in 5K images for training and 5K for testing. For instance segmentation, we employed Pascal SBD 2012 [23], that contains the same set of 20 classes but also reports the instance segmentation annotations. We used the standard split of Pascal SBD 2012, using 8498 images for training and 2857 for evaluation. Following [51], for both object detection and instance segmentation we implement the following experimental protocol: each training step contains all the images that have at least one bounding box of a novel class. We remark that at each training step it is assumed to have only labels for bounding boxes of novel classes, while all the other objects that appear in the image, either belonging to past or future classes, are not annotated. This is a very realistic setup since it does not make any assumption on the objects present in the images and reduces the amount of annotation required in each incremental step.

4.2. Implementation Details

For object detection, we followed previous works [21, 27, 29, 43, 57, 59] and we use the Faster R-CNN architecture with a ResNet-50 backbone. Similarly, for instance segmentation, we employ the Mask R-CNN [24] architecture with ResNet-50 backbone. Both backbones are initialized using the ImageNet pretrained model [11]. We used the same training protocol of [43, 51] but we increased the batch size from 1 to 4 to reduce the time required for training, scaling accordingly the learning rate and number of iterations. In particular, for object detection we train the network with SGD, weight decay 10^{-4} and momentum 0.9. We use an initial learning rate of $4 \cdot 10^{-3}$ for the first learning step and $4 \cdot 10^{-4}$ in the followings. We performed 10K iterations when adding 5 or 10 classes, while we trained for 2.5K when learning only one or two classes. We apply the same data augmentation of [43, 51]. We set λ_2 equal to 0.1, 0.5, and 1 when adding 10 classes, 5, and 1 or 2 classes, respectively. λ_1, λ_3 are set to 1.

4.3. Object Detection Results

As done by previous works [29, 43, 51, 57, 59], for incremental object detection we evaluate our method considering experimental settings adding a different number of classes in one or multiple training steps. We report adding 10 (*10-10*), 5 (*15-5*) or 1 (*19-1*) class in a single incremental step and performing two incremental steps adding 5 classes (*10-5*), five steps adding two classes (*10-2*) and either ten (*10-1*) or five (*15-1*) steps adding one class. As in previous works, we split the classes following the alphabetical order.

Single-step incremental settings (10-10, 15-5, 19-1). Results are reported in Tab. 1. The *Avg* metric equally weights new and old classes averaging their aggregated mAP. We benchmark MMA against previous works reporting the re-

Table 1. mAP@0.5% results on single incremental step on Pascal-VOC 2007. Methods with † come from reimplementation. Methods with * use exemplars.

	19-1				15-5				10-10			
Method	**1-19**	**20**	**1-20**	**Avg**	**1-15**	**16-20**	**1-20**	**Avg**	**1-10**	**11-20**	**1-20**	**Avg**
Joint Training	75.3	73.6	75.2	74.4	76.8	70.4	75.2	73.6	74.7	75.7	75.2	75.2
Fine-tuning	12.0	62.8	14.5	37.4	14.2	59.2	25.4	36.7	9.5	62.5	36.0	36.0
ILOD (Fast R-CNN) [51]	68.5	62.7	68.3	65.6	68.3	58.4	65.9	63.4	63.2	63.1	63.2	63.2
ILOD (Faster R-CNN) [51] †	70.3	65.2	70.0	**67.8**	72.5	58.0	68.9	65.3	69.2	53.0	61.1	61.1
Faster ILOD [43]	68.9	61.1	68.5	65.0	71.6	56.9	67.9	64.3	69.8	54.5	62.1	62.1
Faster ILOD [43] †	70.9	64.3	70.6	67.6	73.5	55.6	69.1	64.6	71.1	52.3	61.7	61.7
PPAS [60]	70.5	53.0	69.2	61.8					63.5	60.0	61.8	61.8
MVC [57]	70.2	60.6	69.7	65.4	69.4	57.9	66.5	63.7	66.2	66.0	66.1	66.1
OREO* [27]	69.4	60.1	68.9	64.7	71.8	58.7	68.5	65.2	60.4	68.8	64.6	64.6
OW-DETR* [21]	70.2	62.0	69.8	66.1	72.2	59.8	69.1	66.0	63.5	67.9	65.7	65.7
ILOD-Meta* [29]	70.9	57.6	70.2	64.2	71.7	55.9	67.8	63.8	68.4	64.3	66.3	66.3
MMA	71.1	63.4	**70.7**	67.2	73.0	60.5	**69.9**	**66.7**	69.3	63.9	**66.6**	**66.6**

Table 2. mAP@0.5% results on multi incremental steps on Pascal-VOC 2007. Methods with † come from reimplementation.

	10-5				10-2				15-1				10-1			
Method	**1-10**	**11-20**	**1-20**	**Avg-S**	**1-10**	**11-20**	**1-20**	**Avg-S**	**1-15**	**16-20**	**1-20**	**Avg-S**	**1-10**	**11-20**	**1-20**	**Avg-S**
Joint Training	74.7	75.7	75.2	75.2	74.7	75.7	75.2	75.2	76.8	70.4	75.2	73.5	74.7	75.7	75.2	75.2
Fine-tuning	6.6	28.3	17.4	21.8	5.2	12.3	8.8	16.7	0.0	8.0	2.4	6.7	0.0	4.6	2.3	8.6
ILOD (Faster R-CNN) [51] †	67.2	59.4	63.3	65.2	62.1	49.8	55.9	62.2	65.6	47.6	60.2	65.8	52.9	41.5	47.2	59.1
Faster ILOD [43] †	68.3	57.9	63.1	65.5	64.2	48.6	56.4	62.8	66.9	44.5	61.3	67.1	53.5	41.0	47.3	60.4
MMA	66.7	61.8	**64.2**	**67.3**	65.0	53.1	**59.1**	**63.8**	68.3	54.3	**64.1**	**67.5**	59.2	48.3	**53.8**	**62.4**

sults on the same settings. We compare either with methods using rehearsal [21,27,29] or not using them [43,51,57,59]. We underline that the former methods are not compared fairly with MMA, since we do not use any replay memory to store old samples. Furthermore, for a fair comparison we report ILOD [51] and Faster ILOD [43] using our same architecture and training protocol. Finally, we report two simple baselines: the joint training upper bound, where the architecture is trained using the whole dataset and all the annotations, and the fine-tuning, where the architecture is trained on the new data using Eq. (1), without employing any regularization strategy.

As can be noted in Tab. 1, fine-tuning suffers a large drop in performance on the old classes, clearly indicating that catastrophic forgetting is an issue to be addressed. While previous works improve the performance, addressing the forgetting issue, MMA outperforms all the previous methods, also the ones that uses exemplars to avoid forgetting, demonstrating the validity of our approach. In particular, when comparing with ILOD [51] and Faster ILOD [43], we note that our method achieve comparable performance on old classes but outperforms them on the new classes, outperforming them of 1% on both 19-1 and 15-5, and even by 10% on the 10-10 setting. We argue that the improvement is largely due to the unbiased distillation loss, that modeling the missing annotations, removes incoherent training objectives, increasing the performance. Comparing MMA to previous state-of-the-art, we note that it outperforms the competitive rehearsal strategies in every setting. On the 19-1 setting, MMA outperforms the ILOD-Meta by 0.5% considering equally every class (*1-20*) and by 1.1% OW-DETR when considering equally old and new classes (*Avg*). Similarly, in the 15-5 and 10-10 settings, MMA outperforms the best rehearsal method by 0.9% and 0.3% on all the classes 0.7% and by 0.3% on the *Avg* metric, respectively.

Multi-step incremental settings (10-5, 10-2, 15-1, 10-1). While performing a single training step is valuable to evaluate the ability to alleviate catastrophic forgetting, a more realistic setting is to perform multiple incremental steps adding new classes. In this section, we analyze the behavior of MMA against three baselines: fine-tuning, ILOD [51], Faster ILOD [43], all implemented following our experimental protocol. We report the results for the four considered settings in Tab. 2, showing the mAP% over multiple incremental steps and Fig. 3, where the results after the last incremental step are reported. Tab. 2 further reports the average performance across multiple steps *Avg-S*.

We can observe that performing multiple incremental steps is challenging and existing methods performances drop badly compared to single step scenarios. In particular, fine-tuning the network on new data, without using any

Table 3. mAP@(0.5,0.95)% results of incremental instance segmentation on Pascal-VOC 2012.

	19-1				15-5			
Method	**1-19**	**20**	**1-20**	**Avg**	**1-15**	**16-20**	**1-20**	**Avg**
Joint Training	40.4	54.1	41.1	47.2	41.0	41.2	41.1	41.1
Fine-tuning	6.7	46.3	8.7	26.5	1.9	35.3	10.2	18.6
Fine-tuning w/ Eq. (3)	12.5	47.5	14.3	30.0	13.0	35.5	18.6	24.2
ILOD [51]	40.1	38.3	40.0	39.2	39.2	30.8	37.1	35.0
Faster ILOD [43]	40.6	38.1	40.4	39.3	39.4	30.3	37.1	34.8
MMA	40.6	43.0	40.8	41.8	38.2	33.7	37.1	35.9
MMA + ℓ_{dist}^{MASK}	41.0	42.8	**41.1**	**41.9**	40.2	32.2	**38.2**	**36.2**

Table 4. Ablation study of the contribution of MMA components in the 15-5 setting. Results are mAP@0.5%. MMA is in green.

Eq. (3)	ℓ_{dist}^{RCN}	ℓ_{dist}^{RPN}	**1-15**	**16-20**	**1-20**	**Avg**
-	-	-	14.2	59.2	25.4	36.7
✓	-	-	40.0	57.8	44.4	48.9
✓	UKD	-	67.3	60.3	65.6	63.8
✓	l2	✓	**73.7**	56.8	69.5	65.3
✓	CE	✓	72.8	59.4	69.5	66.1
✓	UKD	✓	73.0	**60.5**	**69.9**	**66.7**

technique to avoid forgetting, lead to completely forgets the old classes, reaching, in the last step, performances close to 0% on old classes. ILOD [51] and Faster ILOD [43] substantially alleviate catastrophic forgetting, leading to better results both on old and new classes. However, when comparing with MMA, we see that both ILOD and Faster ILOD achieve worse results. In particular, after the last step, it is evident that MMA obtain better performances on novel classes: +2.4% o 10-5, +3.3% on 10-2, +6.3% on 15-1, and 6.8% on 10-1 w.r.t. the best among the baselines. Furthermore, MMA also obtains comparable or greater performance than previous methods on the old classes. Overall, MMA outperforms the best among ILOD and Faster ILOD by 0.9% on 10-5, 2.7% on 10-2, 2.8% on 15-1, and 6.5% on the 10-1 setting. We note that the improvement is larger when adding more classes, indicating that our method is better suited to performing multiple-incremental steps. Considering the trend over multiple training steps in Fig. 3, we note that MMA is always comparable or better than previous methods. In particular, it is remarkable that MMA largely outperforms the other methods when increasing the number of training steps, as shown in the 10-1 setting.

4.4. Instance Segmentation Results

Following the protocol used in incremental object detection, we evaluate our method considering two experimental settings: adding one (*19-1*) and five (*15-5*)classes in a single training step. As in object detection, we follow the alphabetical order of the dataset. Following the standard practice on instance segmentation, we report the mAP averaged across 11 IoU thresholds, ranging from 0.5 to 0.95, with a step of 0.05. We compare MMA with fine-tuning, fine-tuning using the unbiased classification loss (Eq. (3)), ILOD [51] and Faster ILOD [43]. For all the methods we employ the same architecture and hyper-parameters.

Tab. 3 shows the results for the 19-1 and 15-5 settings, reporting the average mAP of new and old classes separately, the average over all classes, and the average of new and old classes (*Avg*), weighting them equally. We can see that fine-tuning shows an impressive forgetting on old classes, both on the 19-1 and 15-5 settings. Introducing the unbiased classification loss (Eq. (3) helps in alleviating forgetting but the results are still low on old classes, clearly indicating that introducing a technique to prevent forgetting is required. ILOD and FasterILOD, in fact, improve the performances on old classes. However, forgetting is prevented at the cost of a decrease in performance on novel classes: they both loses nearly 8% on the 19-1 and 5% on the 15-5 with respect to fine-tuning. Differently, employing our proposed MMA we clearly improve the performance, preventing forgetting while showing good performance on novel classes. In particular, w.r.t. ILOD and Faster ILOD, MMA obtains, on new classes, nearly +5% and +3%, respectively on 19-1 and 15-5, while showing comparable performance on old classes. Considering the extended version of MMA (MMA + ℓ_{dist}^{MASK}), it slightly improves the performance on old classes w.r.t. MMA, while obtaning comparable results on the new ones. Overall, it obtains 41.1% and 38.2% on the 19-1 and 15-5, respectively, 0.3% and 0.9% better than MMA. Interestingly, we note that, without any regularization on the mask head (MMA), we can still achieve good segmentation performance. This is due to the non competitiveness among classes on the mask head, which only regress a binary segmentation mask, while the class is predicted by the classification head, as in standard Faster R-CNN. Overall, MMA and its extension demonstrate to outperform the other baselines in instance segmentation, showing a good trade-off between learning the new classes and avoiding to forget the old ones.

4.5. Ablation Study

In Table 4 we report a detailed analysis of our contributions, considering 15-5 setting in incremental object detection. We ablate each proposed component: the unbiased classification loss (Eq. (3)), the classification head knowledge distillation loss (ℓ_{dist}^{RCN}), the use of the RPN distillation loss (ℓ_{dist}^{RPN}), and finally, the use of a feature distillation loss, as proposed in [43]. The first row indicates fine-tuning the network on the new data, without applying any regularization. It can be noted that the performances are poor on the old classes, while it achieves good performance on the new ones. Adding the unbiased classification, the per-

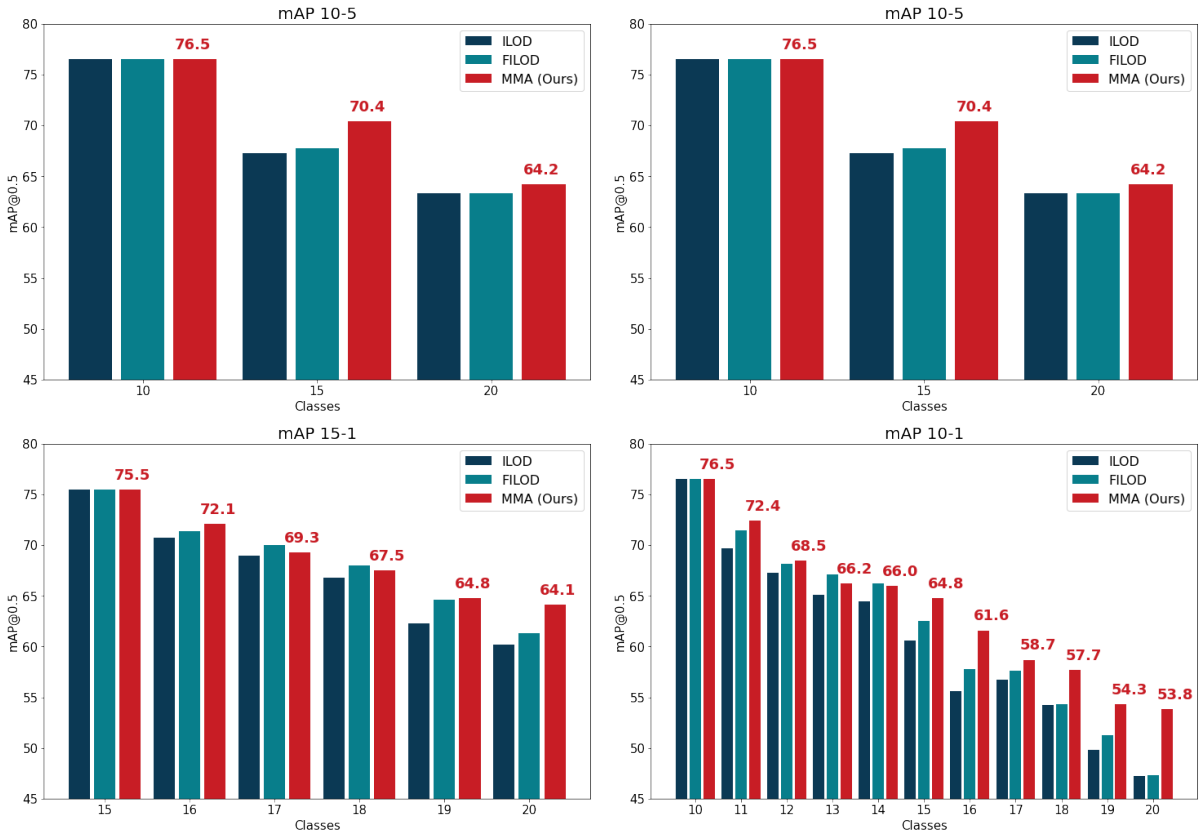

Figure 3. mAP% results on multiple incremental steps on Pascal-VOC 2007.

formance on the old classes substantially improves: from 14.2% to 40.0%. This is due to the handling of missing annotation that alleviates forgetting. Introducing the unbiased distillation loss in Eq. (6) (UKD), the performances improves significantly, both on old classes, reaching 67.3%, and new classes, going from 57.8% to 60.3%. We argue that the performances on the new classes improves thanks to the distillation loss since the model learns to better distinguish the old classes from the new ones, improving the overall precision, We then introduce the RPN distillation loss, obtaining the final MMA model. We see that the performance further improves on old classes, achieving 73.0%, while the performance on the new classes is comparable.

Finally, we compare the unbiased knowledge distillation in MMA with other possible choices. Inspired by previous works we employ the L2 loss on the normalized classification scores [43, 51] and the cross-entropy (CE) loss between the probability of old classes [31]. We see that MMA distillation outperforms them, especially on the new classes, clearly demonstrating that modeling the missing annotations is essential to properly learn them. Overall, MMA achieves on the average of old and new class performance 66.7%, 1.4% and 0.6% more than using the L2 loss or the cross-entropy loss.

5. Conclusions

We studied the incremental learning problem in object detection considering an issue mostly overlooked by previous works. In particular, in each training step only the annotation for the classes to learn is provided, while the other objects are not considered, leading to many missing annotations that mislead the model to predict background on them, exacerbating catastrophic forgetting. We address the missing annotations by revisiting the standard knowledge distillation framework to consider non annotated regions as possibly containing past objects. We show that our approach outperforms all the previous works without using any data from previous training steps on the Pascal-VOC 2007 dataset, considering muliple class-incremental settings. Finally, we provide a simple extension of our method in the instance segmentation task, showing that it outperforms all the baselines. We hope that our work will set a new knowledge distillation formulation for incremental object detection methods. We leave extending our formulation to one-stage detectors as a future work.

References

[1] Manoj Acharya, Tyler L Hayes, and Christopher Kanan. Rodeo: Replay for online object detection. In *BMVC*, 2020. 2

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018. 1, 2

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1, 2

[4] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248, 2018. 2

[5] Fabio Cermelli, Dario Fontanel, Antonio Tavera, Marco Ciccone, and Barbara Caputo. Incremental learning in semantic segmentation from image labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1

[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. 1, 2, 3

[7] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental and weakly-supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1

[8] Fabio Cermelli, Massimiliano Mancini, Yongqin Xian, Zeynep Akata, and Barbara Caputo. Prototype-based incremental few-shot segmentation. In *British Machine Vision Conference (BMVC 2021)*, 2021. 1

[9] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547, 2018. 2

[10] Li Chen, Chunyan Yu, and Lvcai Chen. A new knowledge distillation for incremental object detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2019. 1, 2, 3, 4

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[12] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5138–5146, 2019. 2

[13] 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. 1

[14] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 86–102. Springer, 2020. 1, 2

[15] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. *arXiv preprint arXiv:2111.11326*, 2021. 2

[16] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 2, 5

[17] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 2

[18] Enrico Fini, Stéphane Lathuiliere, Enver Sangineto, Moin Nabi, and Elisa Ricci. Online continual learning under extreme memory constraints. In *European Conference on Computer Vision*, pages 720–735. Springer, 2020. 1, 2

[19] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1, 2, 3

[20] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 2

[21] Akshita Gupta, Sanath Narayan, KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Ow-detr: Open-world detection transformer. *arXiv preprint arXiv:2112.01513*, 2021. 2, 5, 6

[22] Yu Hao, Yanwei Fu, Yu-Gang Jiang, and Qi Tian. An end-to-end architecture for class-incremental object detection with knowledge distillation. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2019. 1, 2, 3, 4

[23] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*, 2011. 5

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 2, 3, 5

[25] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. 2015. 2, 3

[26] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, 2019. 2

[27] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detec-

tion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5830–5840, 2021. 2, 5, 6

[28] 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. 1, 2

[29] Joseph Kj, Jathushan Rajasegaran, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Incremental object detection via meta-learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2021. 2, 5, 6

[30] Dawei Li, Serafettin Tasci, Shalini Ghosh, Jingwen Zhu, Junting Zhang, and Larry Heck. Rilod: Near real-time incremental learning for object detection at the edge, 2019. 2

[31] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 1, 2, 3, 8

[32] 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. 2

[33] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 1

[34] Liyang Liu, Zhanghui Kuang, Yimin Chen, Jing-Hao Xue, Wenming Yang, and Wayne Zhang. Incdet: In defense of elastic weight consolidation for incremental object detection. *IEEE Transactions on Neural Networks and Learning Systems*, 32(6):2306–2319, 2021. 2

[35] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[36] Xialei Liu, Hao Yang, Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Multi-task incremental learning for object detection. *arXiv preprint arXiv:2002.05347*, 2020. 2

[37] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV*, 2018. 2

[38] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, 2018. 2

[39] 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, 2, 3

[40] Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. In *ICCV-WS*, pages 0–0, 2019. 1

[41] Umberto Michieli and Pietro Zanuttigh. Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1114–1124, 2021. 1

[42] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11321–11329, 2019. 2

[43] Can Peng, Kun Zhao, and Brian C. Lovell. Faster ilod: Incremental learning for object detectors based on faster rcnn, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[44] Can Peng, Kun Zhao, Sam Maksoud, Tianren Wang, and Brian C. Lovell. Diode: Dilatable incremental object detection, 2021. 2

[45] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M Hospedales, and Tao Xiang. Incremental few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13846–13855, 2020. 2

[46] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 1, 2

[47] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018. 1, 2

[48] 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. 1, 2, 3

[49] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. 2016. 1, 2

[50] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *arXiv preprint arXiv:1705.08690*, 2017. 2

[51] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *ICCV*, 2017. 1, 2, 3, 4, 5, 6, 7, 8

[52] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 2

[53] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 2

[54] Chenshen Wu, Luis Herranz, Xialei Liu, Joost van de Weijer, Bogdan Raducanu, et al. Memory replay gans: Learning to generate new categories without forgetting. In *NeurIPS*, 2018. 2

[55] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, 2019. 2

[56] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Com-*

puter Vision and Pattern Recognition, pages 3014–3023, 2021. 2

[57] Dongbao Yang, Yu Zhou, and Weiping Wang. Multi-view correlation distillation for incremental object detection, 2021. 2, 5, 6

[58] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017. 2

[59] Wang Zhou, Shiyu Chang, Norma Sosa, Hendrik Hamann, and David Cox. Lifelong object detection. *arXiv preprint arXiv:2009.01129*, 2020. 1, 2, 3, 4, 5, 6

[60] Wang Zhou, Shiyu Chang, Norma Sosa, Hendrik Hamann, and David Cox. Lifelong object detection. *ArXiv*, abs/2009.01129, 2020. 6

[61] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2