Re-examining Distillation for Continual Object Detection

Eli Verwimp[1]
eli.verwimp@kuleuven.be
Kuo Yang[2]
yang.kuo@huawei.com
Sarah Parisot[2]
sarah.parisot@huawei.com
Lanqing Hong[2]
honglanqing@huawei.com
Steven McDonagh[2]
steven.mcdonagh@huawei.com
Eduardo Pérez Pellitero[2]
e.perez.pellitero@huawei.com
Matthias De Lange[1]
matthias.delange@kuleuven.be
Tinne Tuytelaars[1]
tinne.tuytelaars@kuleuven.be

[1] KU Leuven
[2] Huawei Noah's Ark Lab

Abstract

Training models continually to detect and classify objects, from new classes and new domains, remains an open problem. In this work, we conduct an analysis of why and how object detection models forget catastrophically. We focus on distillation-based approaches in two-stage networks; the most-common strategy employed in contemporary continual object detection work. Distillation aims to transfer the knowledge of a model trained on previous tasks -the teacher- to a new model -the student- while it learns the new task. We show forgetting happens mostly in the classification head, where wrong, yet overly confident teacher predictions prevent student models from effective learning. Our analysis provides insights in the effects of using distillation techniques, and serves as a foundation that allows us to propose improvements for existing techniques by detecting incorrect teacher predictions, based on current ground-truth labels, and by employing an adaptive Huber loss as opposed to the mean squared error for the distillation loss in the classification heads.

1 Introduction

Continual learning (CL) is a fast growing research topic within machine learning research. Numerous methods have been proposed to make models learn from non-stationary streams of data and prevent catastrophic forgetting of older knowledge [20, 24, 29]. The challenges of continual learning can be understood in light of the stability-plasticity dilemma [27], observed in both artificial and biological neural networks. We seek to balance the stability of trained models (preserving knowledge acquired during training), with their ability to adapt to new data (plasticity). A crucial challenge is the prevention of catastrophic forgetting [10]. This phenomenon is observed in neural networks whenever there is change in the input data distribution, causing the network to entirely forget how to classify or detect previous data. Designing models and architectures that achieve both stability and plasticity properties remains a difficult task, and often a trade-off has to be made.

Continual learning has been mainly studied in the context of image classification, considering both class-incremental settings [20, 24, 29], where new classes are introduced over time, and domain-incremental settings [18], where the input domain of a class shifts during training. Current popular techniques include revisiting a small set of old images, adding regularization losses to the classification loss and/or freezing parts of the model. Research involving more complex tasks, such as object detection, has been less common [14, 16, 28]. While classification tasks usually aim to recognize a single object per image, detection brings additional challenges. Object detection solutions often combine classification with a class agnostic regression mechanism which has to accurately predict the location of an object, requiring to learn what is an object of interest and what belongs to the background. Detection models should detect all objects that are part of an image, without knowing beforehand whether there will be one, a dozen, or none. Continual object detection (COD) adds another challenge; objects learned during previous COD tasks can appear in subsequent tasks and vice versa, albeit without labels. This is in contrast to classification, where incremental tasks are fully disjoint. These extra challenges and requirements prevent current classification techniques from being applied directly to detection models. In light of these challenges, COD approaches most commonly rely on distillation-based regularization so as to transfer knowledge from a teacher model, trained on one task, to a student model to be trained on the subsequent task [16, 23, 28, 37].

Understanding why catastrophic forgetting occurs and how existing methods alleviate the issue has been investigated to some extent for classification tasks. Knoblauch *et al.* [17] approached the problem from a theoretical point of view, and Benzing [4] provides an analysis and a unifying framework for regularization methods. Nonetheless, to the best of our knowledge, there exists no detailed analysis of COD, the mechanisms in which forgetting occurs, as well as the impact on distillation-type solutions.

In this work, we seek to understand how object detection models forget catastrophically, and more specifically how distillation can alleviate forgetting. Our analysis shows that distillation allows effectively learning a region proposal network (RPN) that performs well on all tasks, but prevents the classification head to adapt properly to new classes, due to overly confident, erroneous, teacher predictions. To address this, we introduce two modifications to the distillation loss, which facilitate inclusion of new classes. Firstly, we rely on task specific bounding box annotations to identify teacher errors and adjust its influence; second, we replace the standard Mean Squared Error (MSE) with a Huber loss, affording more robust knowledge transfer between tasks. We show that our modifications to SOTA distillation based method Faster-ILOD provide consistent improvements, achieving state of the art performance in the class incremental setting.

To summarize, our contributions are the following:

- To the best of our knowledge, we are the first to propose a detailed analysis scheme of continual object detection models' performance, breaking down how individual detector components influence forgetting.[1]
- We provide new insights on the behavior of distillation solutions and address their limitations using two simple modifications, and demonstrate their benefits on a standard distillation-based method and yield state of the art performance on class incremental benchmarks.

2 Related Work

Continual Learning. The taxonomy of Continual Learning benchmarks has become quite extensive over the last few years, we therefore discuss here only essential literature and refer the reader to [7, 26] for a more extensive review. Usually, the distribution of the input data changes at discrete steps and *task* is used to refer to all data between successive steps [7, 33]. When a task contains new classes, this is called class-incremental learning [26, 29]. In domain-incremental learning, the distribution of already present classes changes instead [18]. When only a single epoch is allowed on each task, this is the field of streaming continual learning [3, 6, 12]. For this work, we do allow more than one epoch per task. Note that this implies that the model is aware of when the input data changes. See [33] for an overview of the three scenarios mentioned above. Following [7], existing continual learning methods can be split into three main categories. Regularization methods add extra losses to either keep a model close in parameter space to the final model of the previous task [2, 15], or encourage similar outputs for similar inputs [20]. Replay methods store a small subset of the samples of previous tasks and reuse them during training of new tasks to prevent forgetting [5, 34]. Parameter isolation methods [25, 31] isolate parts of the network and dedicate them to old tasks, but this requires the model to know to which task an image belongs, which is not considered here.

Continual Object Detection Knowledge distillation is a technique first proposed by Hinton *et al.* [13], to transfer knowledge from a large (teacher) to a small (student) network. Li *et al.* [20], adapted the idea and applied it to a continual classification task. Later, Shmelkov *et al.* [32] introduced the idea to COD, after which others took over the concept [16, 23, 28, 37]. Initially, Hinton *et al.* used the Kullback–Leibler (KL) divergence of the output of student and teacher as the loss function. However, all previously mentioned works in COD use an L_2-loss (i.e. mean squared error or MSE), with the exception of [37], who use a smoothed L_1 for the regression outputs, and [16], who use the KL-divergence for the classification heads. These works mainly differ in which exact outputs they distill (*e.g.* logits *vs.* log-likelihood outputs) and some hyper-parameters (*e.g.* ratio of distillation loss to other losses), but all rely predominantly on MSE losses. Besides distillation, rehearsal [5] is a technique often used to reduce forgetting. Both [14] and [16] use a replay memory to continually learn detection tasks. In the former, the memory is used to train an RPN [30] to recognize unknown objects,

[1] https://github.com/VerwimpEli/Continual_Object_Detection

while in the latter it is used to condition the gradient, as a form of meta-learning [9]. Acharya *et al.* employ a basic form of rehearsal, but their focus is on efficient compression of past data [1].

In this work we will focus on how distillation is and can be used to alleviate forgetting and leave the analysis of rehearsal in COD for future work. While one-stage networks have been considered [19, 36], the majority of works use two-stage networks [1, 14, 16, 23, 32, 37]. Therefore, we focus on two-stage networks, in part as this is a pervasive approach in current literature, but also as its granularity enables a relatively simpler analysis. More specifically, we use Faster-ILOD [28] as our prototypical distillation example. This Faster-RCNN [30] based method is a state of the art approach relying solely on distillation. More recent methods build on Faster-ILOD [16, 23, 37], using similar distillation losses, but additionally employing other techniques (*e.g.* rehearsal). These make it more challenging to disentangle the effects of distillation and those of the other techniques, and therefore we base our analysis on Faster-ILOD.

3 Analysis & Method

In this section, we first introduce the COD setting and relevant evaluation protocols. We then elaborate on object detectors' failure modes, and how they can help us understand why and how forgetting occurs. Finally, we dig deeper into why MSE in the ROI-head distillation fails and propose improvements to cope with these issues.

3.1 Problem Formulation

A standard COD problem is defined as a sequence of object detection tasks, where each task is associated with a set of training images and corresponding bounding box annotations. For now, we focus our analysis on the class incremental COD setting, which is currently the most popular set-up on which methods are developed and evaluated. In this setting, each task comprises a set of unique classes for which annotations are available, effectively separating class annotations into disjoint subsets. An important nuance with respect to standard task incremental classification settings is that an image can be part of multiple tasks, while bounding box annotations cannot. Indeed, an image will be part of a task if it possesses at least one object from the classes associated with this task. Standard COD benchmarks often consist of two tasks, and are referred to as $n_1 + n_2$, where n_1 and n_2 are the number of classes in tasks 1 and 2 respectively. This setting proves complex enough to reveal real-world COD challenges, and yet remains a feasible test-bed for investigative model training and analysis. In the remainder of this section, we first discuss standard COD evaluation and its shortcoming, then detail our analysis process towards understanding how forgetting occurs in COD tasks. For the purpose of our analysis, we use the VOC2007 $10+10$ benchmark [8, 32], consistent with a majority of previous work. This benchmark separates the highly popular PASCAL VOC 2007 object detection dataset into two tasks. The first task (T1) contains the first (alphabetically ordered) ten classes, and the second task (T2) the next ten.

3.2 Disentangling Forgetting in COD

We now detail our analysis process towards the role of distillation in two-stage COD methods and their shortcomings. Two-stage object detectors consist of three main components: a backbone that extracts semantic features, an RPN proposing regions that likely contain an object and the ROI-head, responsible for classifying said regions. See [30] and [11] for more details. We rely on Faster-ILOD to evaluate the impact of distillation, so we briefly detail its workings here. Distillation losses are applied on the three main components of the object detector. The backbone is distilled using the MSE between the features of the teacher's backbone and those of the student. Likewise, for the RPN, an MSE-loss is used on *all* proposals of both RPN's, both for the objectness scores and the regression outputs. To distill the ROI-heads, first 64 out of the 128 proposals with the highest objectness score proposed by the student's RPN are randomly selected, then they are evaluated both by the heads of the teacher and student, and used in a third MSE component. For further details, see [28].

3.2.1 Backbone and RPN

Initial experiments on VOC10+10 indicate that most of the forgetting happens in the ROI-head. Compared to the Faster-ILOD benchmark, freezing the backbone leads to a 1 mAP point decreases, and not employing distillation (i.e. fine-tuning) to a 0.1 mAP decrease, indicating that forgetting isn't crucially happening in the backbone. Using Faster-ILOD's MSE-distillation the RPN respectively finds 98.6% and 98.4% of the labeled objects at IOU-level 0.5, which shows that using distillation forgetting in the RPN can be mostly prevented. The effectiveness of distillation in these tasks is interesting observation, and hints at a fundamental difference between a classification task and a regression task such as bounding box detection. This is possibly explained by the domain-incremental nature of the regression task in contrast to the class-incremental classification task [33]. Given these observations, for the remainder of this paper, we continue with an analysis of the ROI-head and how its continual learning performance can be improved. See Supplementary for details on these preliminary experiments.

3.2.2 ROI-Head

The classification head can either classify candidate boxes as background, or assign one of the classes of interest. When learning a new task without a specific continual learning mechanism, the classification head will only see annotations of new classes, which results in a bias towards those classes. Due to the nature of the content found in the considered benchmark datasets; instances of classes, belonging to previous tasks ("old classes"), will be unannotated in new task images. Yet crucially, such instances may still be present in the samples of the new task (*i.e.* their appearance is now marked as background). The crux of the problem involves classifying these old classes as background; rather than their actual label. As a result, a crucial task for COD models is to preserve identification of classes pertaining to previous tasks in the classification head.

Figure 1 shows the mean average precision (mAP) of three models: one that's only trained on T1, one that's sequentially fine-tuned without any continual learning strategy on T2, and one that's trained as in Faster-ILOD. The results are normalized w.r.t. a joint model that

	← T1 Objects										T2 Objects →											
	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	T1	T2
Only T1	0.96	0.97	0.97	0.99	1.03	0.90	0.99	0.96	1.04	0.77	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.96	0.00
Finetune	0.34	0.02	0.30	0.15	0.03	0.05	0.01	0.15	0.04	0.01	0.69	0.86	1.53	0.97	0.95	0.71	0.79	0.73	0.98	0.87	0.11	0.91
Faster-ILOD	0.91	0.92	1.00	1.10	1.05	0.89	0.96	0.94	1.14	0.90	0.62	0.90	1.23	0.88	0.80	0.56	0.90	0.76	0.74	0.60	0.98	0.80
Ours	0.93	0.96	0.95	1.00	0.99	0.88	0.97	0.93	1.00	0.69	0.89	0.92	1.50	0.98	0.93	0.71	0.95	0.93	0.87	0.89	0.93	0.96

Figure 1: Normalized mAP of the classes in VOC after only learning T1, after finetuning on T2 and after learning T2 with Faster-ILOD and the final average per task. Ours is our improved version of Faster-ILOD, see Sections 3.3 and 4 for further details. The results are normalized as $\frac{r_i}{r_{i,j}}$ with r_i the mAP of class i trained sequentially and $r_{i,j}$ the mAP of class i of a model trained jointly on T1 and T2.

is trained on all classes at once, which is commonly constituted as an upper bound in CL-settings. Unlike the RPN, the ROI-head forgets all but two classes catastrophically, indicating the need for continual learning techniques. The distillation used in Faster-ILOD is effective: the second task is learned well, while forgetting of the former task is prevented. Yet, for the classes of T2, the model using distillation only reaches an average of 80% of the performance of a joint model. The individual results for the classes that make up the second task show that the errors aren't evenly distributed: some classes are learned as well or better than with a joint model, while classes like *table*, *plant* and *tv* only reach about 60%.

To improve upon the Faster-ILOD score, it is essential to understand why some of the classes are not properly learned. From all detected objects of T2, the Faster-ILOD model classifies 76.2% correctly, assigns 17.9% to the wrong class and marks 5.9% as background, see Figure 2. While the number of proposals mistakenly classified as background is higher than in the first task, the wrongly classified samples make up the majority of the errors, hence our focus will be on those. Of all the wrongly classified proposals, 84% are classified as a T1 class and only 16% gets mixed up with another T2 class, see Supplementary for the full confusion matrices. This indicates that the model is overly static, which we conjecture is caused by the effects of the distillation loss. In the following section, we propose to relax this objective towards mitigating this major source of error.

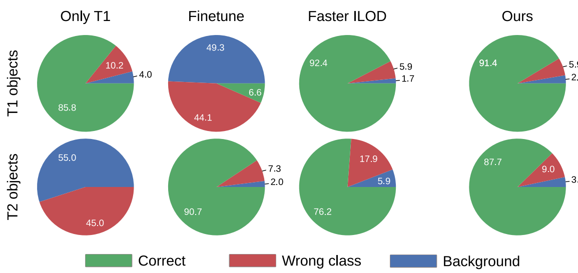

Figure 2: Partitioning of the classification of all region proposals, by a model trained on only the classes of T1, one that is subsequently also finetuned on T2, a Faster-ILOD model and our improvement, see sections 3.3 and 4.

	table	dog	horse	motor	person	plant	sheep	sofa	train	tv
chair co-occurrence	0.7	0.0	0.0	0.0	1.4	0.3	0.0	0.2	0.0	0.2
median bkg. score	7.4	0.9	1.8	2.9	7.0	7.8	1.5	4.6	3.3	5.1

Table 1: Average number of objects in T2 that are in an image with a chair (T1 object) and the median background score of their ground truth bounding box after T1

3.3 Improving ROI-Distillation

During training on earlier tasks, RPNs already detect objects of future tasks even though they aren't labeled yet (see Supplementary). As a result, the teacher will learn to confidently classify these objects as earlier classes or background, which leads to conflicting losses when combining distillation with the cross-entropy classification loss of the ROI-heads. To illustrate this effect, we show four high scoring region proposals of the RPN after the first task (T1) of VOC10+10 in Figure 3. In Faster ILOD, all four are candidates to use in the distillation loss. We claim that these wrongly classified proposals hinder the learning of the T2 classes, since they overlap more with a T2 object (*e.g.* table and dog) than one from T1 (*e.g.* chair). Next, we uncover the issues caused by these proposals further and propose two techniques to alleviate them.

3.3.1 Selective Distillation

Ideally, only proposals that contain old objects are used in the distillation of the ROI-heads. To filter out those that do contain new objects, the IOU of the proposed regions with the current ground truths offers an important clue. If they are above a threshold (we use 0.5), it is likely that these regions primarily contain a new object. Because the teacher has never seen the correct label for these, it can not possibly classify these accurately, and its classification should not be trusted. Therefore, before applying distillation, all proposals with IOU higher than 0.5 with any ground truth are filtered out, and only the others are used for distillation. In Figure 3, this would preclude proposals A and C from being used in the distillation, but not B and D.

3.3.2 Huber Loss

The IOU of proposals B and D in Figure 3 with any old object's ground truth bounding box is below 0.5, but they still contain a significant portion of a new object. Ideally, these should not be used for distillation, but lowering the threshold would mean throwing away too many correct proposals with objects of old tasks. This is problematic, since proposals with partly new objects can still lead to conflicting loss signals. In such cases, the increase in distillation

Figure 3: Four examples of region proposals after training only the first VOC10+10 task. Values shown are IOUs with the ground truth bounding box of the table and the dog, and the maximal logit of the ROI-head prediction of that proposal.

loss should be similar to the decrease in CE-loss. Unfortunately, this is not the case: when using MSE, the distillation loss can become large, especially for objects that co-occur often with objects of the previous task. Since the RPN already detected these co-occurring objects during the first task, the model has learned to classify them confidently as background or as a wrong class, see Table 1 and the Supplementary. This leads to higher background scores for objects that were often in the past task, compared to those that co-occur less often with past objects. An exception to this observation is the person class, which co-occurs often with the chair and has a high background score, yet still scores well. This is most likely due to the over-representation of this class, which constitutes 60% of the new task objects.

The higher initial logits of these objects mean their logits have to change more in order to get a correct classification, which eventually leads to a problem with the MSE-loss. If the distillation loss term of proposals like B and D increases faster than the decrease in classification loss of proposals like A and C, the regularization will dominate. Since MSE is quadratic, this gets worse the higher the initial logits are. If the L_1 distance between the closest correct logits y_n and the initial ones y_o does not grow too large, the MSE is still of the same order of magnitude as the CE-loss. However, for proposals like D this distance is much larger, leading to MSE-losses that blow up and over-regularization.

To alleviate this issue, we propose to change the MSE-loss to a Huber loss for the distillation of the ROI-heads, defined as:

$$\mathcal{L}_h(x,y) = \begin{cases} \frac{1}{2}(x-y)^2 & \text{if } |x-y| < \delta \\ \delta(|x-y| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \quad (1)$$

This is quadratic for differences less than δ and linear for values greater than or equal to δ. For larger differences between correct and initial logits, this ensures that the increase in distillation loss remains of the same order of magnitude as the decrease in CE-loss. In contrast, there isn't a large difference between MSE- and Huber loss for the distillation of old objects, because the difference of their logits will be smaller given that there is no conflicting CE-loss. This leads to the sought after trade-off: more modest distillation for proposals that are close to new objects and require changes in their classification, while remaining strict for other proposals. Adapting MSE losses to a Huber loss has been shown effective in other settings too, see for instance [22].

4 Results

4.1 Benchmarks and evaluation

We evaluate our method on three two-task scenario's of VOC2007 [8] ($10+10$, $15+5$ and $19+1$) and a $40+40$ scenario in MS COCO [21]. We primarily compare our improvements to the original version of Faster-ILOD [28], showing how these fixes improve the distillation in Faster-ILOD, which is our main objective. For context, we then provide SOTA in COD that uses more complex mechanisms; Open World Detection (ORE) [14] and Meta [16]. Both of these methods use a replay memory, with respectively 50 and 10 images per class. In Continual Learning, replay and regularization (distillation) are considered complementing ideas [7], and studying the impact of rehearsal, and its interaction with distillation in COD

	10+10					15+5					19+1				
	First 10	FI	Ours	ORE	Meta	First 15	FI	Ours	ORE	Meta	First 19	FI	Ours	ORE	Meta
T1	68.7	69.8	66.2 (-3.6)	60.4	68.3	74.2	71.6	73.3 (+1.7)	71.8	71.7	73.2	68.9	72.8 (+3.9)	67.9	70.9
T2	-	54.5	64.7 (+10.2)	68.8	64.3	-	56.9	58.2 (+1.3)	58.7	55.9	-	61.1	62.8 (+1.7)	60.1	57.6
Task Average	-	62.1	**65.5 (+3.4)**	64.6	**66.3**	-	64.3	**65.7 (+1.4)**	65.2	63.8	-	65.0	**67.8 (+2.8)**	64.0	64.2
mAP	-	62.1	**65.5 (+3.4)**	64.6	**66.3**	-	67.9	**69.5 (+1.6)**	68.5	67.8	-	68.5	**72.3 (+3.8)**	67.5	70.2

Table 2: Results on three two-task VOC benchmarks, 10+10, 15+5 and 19+1. The results are VOC mAP @IOU 0.5 and tested on the full test set, as explained in Section 3.1. We compare ours to Faster-ILOD [28] (FI), the values in brackets show the difference. Additionally, we compare to SOTA methods ORE [14] and Meta [16], which both use a replay memory. The initial mAP is in the column first X. For each method the mAP of the first (T1) and second (T2) task is shown, as well as the overall mAP and the average mAP of both tasks

constitutes future work. The results for Faster-ILOD, ORE and Meta are from [16], and publicly available T1 models of each method are used to allow a fair comparison. We show the mAP results for each task (T1 and T2), the mAP over all classes and the average of both tasks. Reporting only the final mAP over all classes may hide differences across methods and tasks, especially in the 19+1 benchmark where the single new class does not influence the overall mAP much. For VOC we report the mAP at IOU 0.5, for MS COCO the average mAP at IOUs from 0.5 to 0.95 with step-size 0.05, which both are the respective defaults.

4.2 Implementation details

We use a standard Faster RCNN model, with ResNet-50 as backbone, which is the same setting as in the methods we compare to. For all benchmarks, we use SGD with momentum 0.9. In the 10+10 benchmark, the new task is trained for 20k iterations, batch size 4 and learning rate 0.001, decaying after 15k and 17.5k iterations. 15+5 is trained for 10k iterations, batch size 4, learning rate 0.001, decaying at iteration 7.5k. The 19+1 setting is trained with batch size 1, 10k iterations and learning rate 0.0001. The MS Coco benchmark is trained for 90k iterations, 16 images per batch and initial learning rate 0.02. These settings were chosen to be similar to both [28] and [16], see Supplementary for a full comparison. The implementation is written in the Detectron2 framework [35], and is publicly available.

4.3 Results and Analysis

Tables 2 and 3 show the results for VOC and COCO. Our adaptation consistently improves over the original Faster-ILOD on all three benchmarks, with an increase of 3.4, 1.6 and 3.8 mAP for the 10+10, 15+5 and 19+1 setting respectively. For the latter two, our adaptation even does better than both SOTA methods, without relying on a rehearsal memory. Compared to Faster-ILOD our method is able to adapt better to the new task, without compromising much on the old task. The same is true for the results on COCO, which improves over Meta and Faster-ILOD. To verify that our adaptations resolved the issues we exposed, we included *Ours* in Figures 1 and 2. The mAPs of the classes that weren't properly learned at first are markedly higher than in the original Faster ILOD. The number of classification mistakes on the second task nearly halved, while only compromising 2% in the mistakes of the first task. Table 4 shows an ablation of the VOC10+10 task, which shows that both the Selective Distillation and the Huber loss have a positive influence, and combining both leads to the best results.

	AP	AP_{50}	AP_{75}
Joint	31.5	50.9	33.5
Faster-ILOD	20.6	40.1	-
Meta	23.8	40.5	24.4
Ours	**26.4**	**44.3**	**27.4**

	T1	T2	mAP
Faster ILOD	69.8	54.5	62.1
Huber	**71.2**	58.0	64.6
Selective Distillation	69.5	59.7	64.6
Both	66.2	**64.7**	**65.5**

Table 3: *(Left)* AP, AP50 and AP75 on the minival of MS COCO40+40. AP is the average of the APs at IOU levels from 0.5 to 0.95 with steps of 0.05. Results on the full validation set and per task are in the Supplementary.

Table 4: *(Right)* Ablation study of the VOC10+10 task with Selective Distillation and/or Huberloss. Additional ablations are available in the Supplementary.

5 Limitations

We identify a number of limitations in our study that can lead to further investigation. Namely; all three datasets used include images with annotations for *both* tasks, albeit unlabeled. Absence of such images may lead to different results. Furthermore, we have endeavored to account for all mechanisms that contribute to both the aspects of (1) forgetting; and (2) prevention of new-task learning, however the inherent complexity of object detectors dictates that additional causal factors may yet be uncovered. Finally we note that our class-incremental experimental work currently considers only two-task benchmarks. Extensions to longer task sequences will enable insight into distillation mechanisms in such additional challenging scenarios.

6 Conclusion

We undertook a thorough analysis of how models in COD forget and how distillation methods, such as Faster-ILOD, limit forgetting. We identified issues that prevent Faster-ILOD from learning new tasks and propose solutions to fix these problems. Our modifications provide improvement to the current SOTA, on multiple datasets and benchmarks.

7 Acknowledgment

We gratefully acknowledge the support of MindSpore, CANN (Compute Architecture for Neural Networks) and Ascend AI Processor used for this research.

References

[1] Manoj Acharya, Tyler L Hayes, and Christopher Kanan. Rodeo: Replay for online object detection. *arXiv preprint arXiv:2008.06439*, 2020.

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018.

[3] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems*, pages 11816–11825, 2019.

[4] Frederik Benzing. Understanding regularisation methods for continual learning. *arXiv e-prints*, pages arXiv–2006, 2020.

[5] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. Continual learning with tiny episodic memories. *arXiv preprint arXiv:1902.10486*, 2019.

[6] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8250–8259, October 2021.

[7] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[8] 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.

[9] Sebastian Flennerhag, Andrei A Rusu, Razvan Pascanu, Francesco Visin, Hujun Yin, and Raia Hadsell. Meta-learning with warped gradient descent. *arXiv preprint arXiv:1909.00025*, 2019.

[10] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

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

[12] Tyler L Hayes, Nathan D Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9769–9776. IEEE, 2019.

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

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

[15] 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.

[16] Joseph Kj, Jathushan Rajasegaran, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Incremental object detection via meta-learning. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–1, 2021.

[17] Jeremias Knoblauch, Hisham Husain, and Tom Diethe. Optimal continual learning has perfect memory and is np-hard. In *International Conference on Machine Learning*, pages 5327–5337. PMLR, 2020.

[18] Matthias Lenga, Heinrich Schulz, and Axel Saalbach. Continual learning for domain adaptation in chest x-ray classification. In *Medical Imaging with Deep Learning*, pages 413–423. PMLR, 2020.

[19] 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. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 113–126, 2019.

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

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

[22] 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, 2020.

[23] Xialei Liu, Hao Yang, Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Continual universal object detection. *arXiv preprint arXiv:2002.05347*, 2020.

[24] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30:6467–6476, 2017.

[25] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

[26] Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification. *arXiv preprint arXiv:2010.15277*, 2020.

[27] Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in psychology*, 4:504, 2013.

[28] 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.

[29] 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.

[30] 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*, 28:91–99, 2015.

[31] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pages 4548–4557. PMLR, 2018.

[32] 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.

[33] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

[34] 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 (ICCV)*, pages 9385–9394, October 2021.

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

[36] Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C-C Jay Kuo. Class-incremental learning via deep model consolidation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1131–1140, 2020.

[37] Wang Zhou, Shiyu Chang, Norma Sosa, Hendrik Hamann, and David Cox. Lifelong object detection. *arXiv preprint arXiv:2009.01129*, 2020.