One-Shot Replay: Boosting Incremental Object Detection via Retrospecting One Object

Dongbao Yang[1,2], Yu Zhou[1,2*], Xiaopeng Hong[3], Aoting Zhang[1,2†], Weiping Wang[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences
[2] School of Cyber Security, University of Chinese Academy of Sciences
[3] Harbin Institute of Technology
{yangdongbao, zhouyu, zhangaoting, wangweiping}@iie.ac.cn, hongxiaopeng@ieee.org

Abstract

Modern object detectors are ill-equipped to incrementally learn new emerging object classes over time due to the well-known phenomenon of catastrophic forgetting. Due to data privacy or limited storage, few or no images of the old data can be stored for replay. In this paper, we design a novel One-Shot Replay (OSR) method for incremental object detection, which is an augmentation-based method. Rather than storing original images, only one object-level sample for each old class is stored to reduce memory usage significantly, and we find that copy-paste is a harmonious way to replay for incremental object detection. In the incremental learning procedure, diverse augmented samples with co-occurrence of old and new objects to existing training data are generated. To introduce more variants for objects of old classes, we propose two augmentation modules. The object augmentation module aims to enhance the ability of the detector to perceive potential unknown objects. The feature augmentation module explores the relations between old and new classes and augments the feature space via analogy. Extensive experimental results on VOC2007 and COCO demonstrate that OSR can outperform the state-of-the-art incremental object detection methods without using extra wild data.

Introduction

Modern object detection methods based on deep learning have achieved remarkable progress, which are usually trained on pre-defined datasets with a fixed number of classes. However, in many practical applications, new object classes often emerge after the detectors have been trained. It is well-known that naive fine-tuning on new classes suffers from catastrophic forgetting (French 1999; Goodfellow et al. 2013; McCloskey and Cohen 1989), severely degrading the performance on old classes (Kirkpatrick et al. 2017). Due to data privacy and limited storage of the devices, few or even no old data are available for training the detectors from scratch. In addition, even if the old data are available, jointly training with both old and new data will take a long training time. Therefore, it is necessary to improve the ability of object detectors to continuously learn new object classes on new data, which is called Incremental Object Detection (IOD).

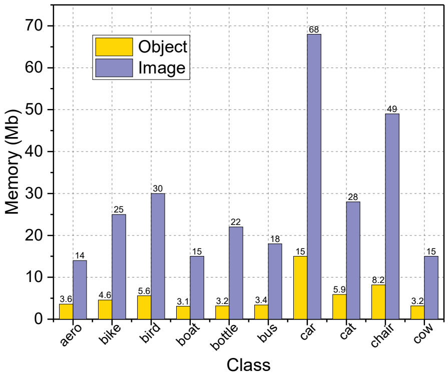

Figure 1: Memory usage (Mb) of all images and cropped objects for the first 10 classes on VOC2007.

According to the ways of tackling catastrophic forgetting, incremental learning methods can be divided into: regularization-based, replay (or memory)-based and parameter-isolation-based (Delange et al. 2021). Recent incremental object detection methods (Chen, Yu, and Chen 2019; Hao, Fu, and Jiang 2019; Hao et al. 2019; Li et al. 2019; Shmelkov, Schmid, and Alahari 2017; Zhang et al. 2020; Zhou et al. 2020; Yang et al. 2022a,b) mainly resort to designing complex regularization-based methods, which focus on knowledge preservation and utilize distillation techniques to transfer the knowledge learned from the old data to the new model.

To this day, replay methods are not thoroughly studied in recent incremental object detection methods. For the classification task, traditional replay methods commonly save a set of original training samples for old classes and combine it with the new training set in the incremental learning procedure. Previous methods have proven that data replay can boost performance and mitigate catastrophic forgetting (Rebuffi et al. 2017). However, it still has limitations if traditional replay methods are straightforwardly transferred to the detection task. On one hand, due to the limited memory on devices, the number of large-size samples to be stored is restricted. On the other hand, due to data privacy, completed

[*] Corresponding author

[†] Aoting Zhang's work is cooperated and partially sponsored by Peking University.

forms of exemplars like the whole image and the bounding-box annotations may be inaccessible. Therefore, how to represent the memory space efficiently with marginal storage and labeling costs is crucial for replay-based incremental object detection.

Heuristically, to reduce the memory size, we can crop the objects with bounding boxes from the old data instead of storing the original training images. To verify its effect, we calculate the memory usage of images and cropped objects. As shown in Figure 1, for the first 10 classes from VOC2007 according to the alphabetic order, the images take up about 3.9∼6.9 times more memory than cropped objects. It demonstrates that storing objects rather than images can reduce the memory size by a large margin. In this paper, we propose a One-Shot Replay (OSR) method to store only one cropped object for each old class, which utilizes copy-paste as a harmonious design to replay for incremental object detection. Compared with the traditional replay methods, it can reduce the memory size, require no manual bounding-box annotations and decrease the training time consumed on old samples. To increase data diversity and co-occurrence frequency of old and new objects, we design two augmentation modules, including object augmentation and feature augmentation. Compared with the recent regularization-based incremental object detection methods, OSR can reduce the computation cost and improve the variousness of the training samples in the incremental learning procedure.

The contributions of our work are as follows:

- To the best of our knowledge, this is the first attempt to explore the unique data replay method for incremental object detection. We propose the one-shot replay method using copy-paste to augment data, which is a harmonious design for incremental object detection.
- Two sophisticated augmentation modules are specially designed to increase data diversity. Object augmentation is introduced to extend the perception of the potential objects. Feature augmentation is proposed to use the instance-level representation of new classes to enrich the distribution of old classes.
- Extensive experiments on VOC2007 (Everingham et al. 2010) and COCO (Lin et al. 2014) demonstrate the effectiveness of one-shot replay. Compared with traditional replay, it can reduce memory usage and achieve high precision simultaneously.

Related Work

Incremental learning/Continual learning: According to the ways of tackling forgetting, the methods can be categorized into: regularization-based, replay (or memory)-based and parameter-isolation-based (Delange et al. 2021). Regularization-based methods (Kirkpatrick et al. 2017; Aljundi et al. 2018; Li and Hoiem 2017; Liu et al. 2022) introduce additional regularization terms in the loss function, preserving previously learned knowledge when learning on data of new classes. It avoids storing old data and alleviates memory requirements. Replay-based methods (Rebuffi et al. 2017; Shin et al. 2017) store the raw samples or generate pseudo-samples with a generative model. Parameter-isolation-based methods (Aljundi, Chakravarty, and Tuytelaars 2017) allocate different parameters to each task, which can be divided into fixed and dynamic architectures (Mai et al. 2021). Moreover, Zhu et al. (Zhu et al. 2021) propose a dual augmentation method for class incremental learning. Recently, the prompting-based method (Wang, Huang, and Hong 2022) is also proposed for domain incremental learning.

Incremental Object Detection: Shmelkov et al. (Shmelkov, Schmid, and Alahari 2017) propose the first Fast R-CNN (Girshick 2015) based incremental object detection method, which uses EdgeBoxes (Zitnick and Dollár 2014) and MCG (Arbeláez et al. 2014) to precompute proposals. Knowledge distillation is applied to regularize the outputs of the classification and regression layers in the detection head to preserve the performance on old classes. Hao et al. (Hao et al. 2019) freeze RPN to preserve the learned knowledge from the old classes and minimize the difference between the features of the old and new models using a feature-changing loss. They also introduce a new dataset (TGFS) in (Hao, Fu, and Jiang 2019), which is a hierarchical large-scale retail object detection dataset. They propose to use an exemplar set with a fixed size of old data for incremental object detection. Chen et al. (Chen, Yu, and Chen 2019) propose to use a hint loss (L2 loss) to minimize the distance between the features of the old and new models. Li et al. (Li et al. 2019) propose to extract three types of knowledge from RetinaNet (Lin et al. 2017), and they use smooth L1 loss to penalize the feature difference. Zhang et al. (Zhang et al. 2020) pre-train a new model only for the new classes and use a dual distillation function for incremental learning from two teacher models simultaneously.

The effectiveness of storing a few examples for boosting the performance on old classes has been demonstrated in the related works (Hao, Fu, and Jiang 2019; Li et al. 2019; Joseph et al. 2021b), as well as the works in few-shot object detection by Wang et al. (Wang et al. 2020) and open-world object detection by Joseph et al. (Joseph et al. 2021a). However, they just adopt the typical way to store a small balanced set selected from the original training exemplars with the annotations, which does not consider the characteristics of object detection in the incremental learning procedure.

Copy-Paste: Recently, copy-paste as a data augmentation method has been found to be effective for both object detection (Dwibedi, Misra, and Hebert 2017; Kisantal et al. 2019) and instance segmentation (Fang et al. 2019; Ghiasi et al. 2021; Xu et al. 2021). Dwibedi et al. (Dwibedi, Misra, and Hebert 2017) improve detection by the simple cut-and-paste method, which uses the extra instances with annotated masks. Ghiasi et al. (Ghiasi et al. 2021) find that randomly pasting objects can provide solid gains for instance segmentation. However, copy-paste has not been studied as a data replay method in incremental learning for object detection. Since it is an effective method that can handle the scarce data problem for object detection and instance segmentation, we explore copy-paste for data replay in the incremental object detection domain to save memory usage, reduce computation costs and improve performance.

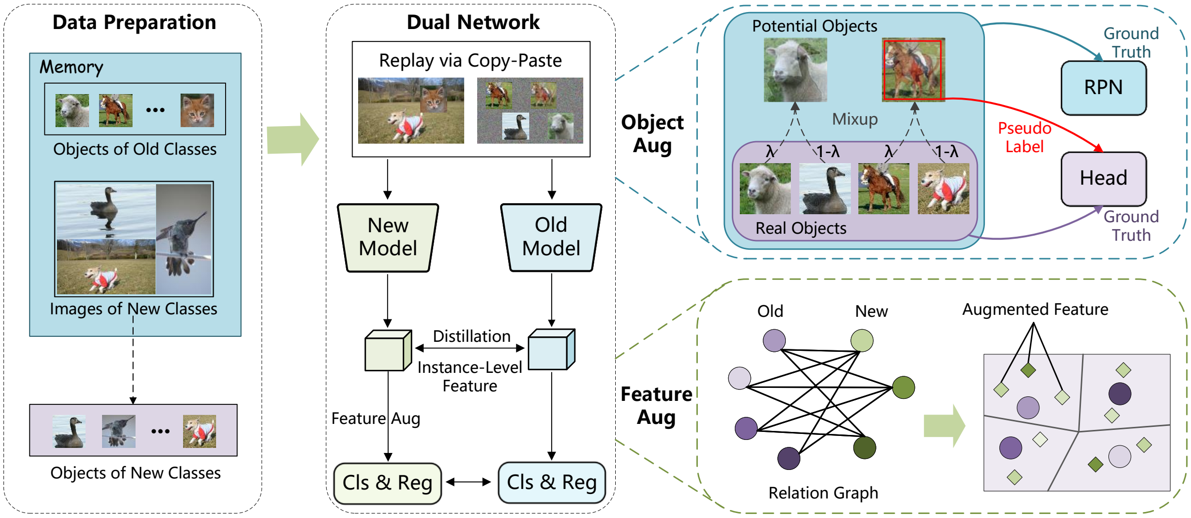

Figure 2: The whole framework of the proposed incremental object detection method based on one-shot replay.

Method

Overview

In this paper, we propose a replay-based incremental object detection method via augmentation, which is based on a dual network, as shown in Figure 2. The frozen old model trained on sufficient old data provides the learned knowledge of old classes. The new model is initialized by the old model and adapted to incrementally learn from the synthetic new data. In each forward step of the incremental learning procedure, we firstly generate synthetic samples with co-occurring objects of old and new classes by randomly copying the stored one-shot object of each old class and pasting them on new samples or clean background. Then, the synthetic samples are input into the dual network, where the old model assists in providing the knowledge of old classes in the features and outputs. To better preserve the learned feature representations of old classes and maintain the discrimination between all classes simultaneously, we enforce the instance-level features and output logits of old classes from the new model to imitate those from the old model.

One-Shot Replay

To minimize the storage of old data, we propose to store only one object for each old class. To fully use the stored objects and increase the diversity of training data, we exploit copy-paste to perform replay for incremental learning, which replays objects of old classes by augmenting new samples. Intuitively, the advantages of using copy-paste for replay include: (1) the object-level samples are easily obtained, which can be cropped from the old data. If the old data are inaccessible, they can also be collected from the wild data; (2) the memory usage of storing object-level samples is far less than storing the whole images; (3) copy-paste can be seen as a kind of augmentation to existing samples which can increase the data diversity; (4) the size of the new training dataset is not changed, which will not increase the number of forward steps and the training time.

The heuristic idea is to copy the cropped objects and paste them into the new training samples, which will not change the size of the dataset, as well as save the training time on extra old samples. It can also increase the co-occurrence frequency of the objects belonging to both old and new classes to strengthen inter-class discrimination. The main steps are as follows: firstly, we randomly select a cropped object from memory and resize it with random width and height in a range. Then, search a position in the new sample for pasting the object, where the IoUs between the object and the ground truths of the new sample should be lower than a threshold. To ensure the search time is within a range, we also define a limited running time to restrict the search process. Finally, new synthetic samples with pasted objects of old classes are generated, and the integrated annotations of the original ground truths and the pasted objects are also available.

Since randomly selected objects of old classes may be deformed or occluded, it is hard to be recognized without contextual information. To ensure the quality of the stored objects, we use the pretrained ResNet-50 (He et al. 2016) on ImageNet to select the representative objects, which are near the mean of objects for each class, and they can be recognized with high confidence.

Object Augmentation

To enhance the ability of the detector to perceive potential objects and improve the recall of candidate proposals, we propose an object augmentation scheme based on Mixup (Zhang et al. 2018). We randomly interpolate two objects (x_a and x_b) from old and new classes to generate a new object and paste it onto the training data, as written in Eq. 1:

$$x_{mix} = \lambda x_a + (1 - \lambda) x_b, \tag{1}$$

where the interpolation coefficient is $\lambda \sim \beta(1.0, 1.0)$.

To reduce the corruption of original new training data caused by copy-paste, we replay these augmented objects on clean background. Moreover, in some cases, the new emerging categories contain only a small number of samples for the incremental learning stage. For example, the number of samples containing “tv monitor” is just 279 on VOC2007. The small size of new samples means a lack of context information for distinguishing background and foreground, new classes and old classes, which will cause the intra-class and inter-class confusion in incremental learning. In addition, it is difficult for the new model to thoroughly learn the differences between the old and new categories with only a few samples, and the new model tends to overfit the new classes. Therefore, we also paste objects of old and new classes on the clean background as a new training sample, compensating for the existing new training data. The synthetic samples are dynamically generated in the training procedure without taking up any memory. This replay method can generate diverse images by randomly combining different objects, increasing the number and diversity of new training samples, as well as preventing overfitting.

After generating the new synthetic samples, there are three kinds of objects in the training data: objects with ground truth (original and pasted objects), objects without annotations (old objects in the new images that are not labeled) and mixed objects. Since the mixed objects are dissimilar to the objects in old data, we only use them for training RPN, which will not damage the learned knowledge of old classes in the detection head. Since the ratio λ of two samples for mixing is randomly sampled, some objects of old classes are still recognizable after being mixed, which can be used for incremental learning. Therefore, we use pseudo labels obtained from the old model to get the annotations of the recognizable objects of old classes, which can increase the diversity of the objects of old classes.

Feature Augmentation

To compensate for the insufficient representation of old classes in feature space, we propose a feature augmentation module. Inspired by (Wang et al. 2021) and (Zhu et al. 2021), the distribution information (class mean μ and covariance Σ) of old classes can be used to augment the feature space and regularize the learning of the classifier. However, limited variations are contained in the distribution information due to the lack of objects of old classes. Since old and new classes may have similar appearances, such as “bus” vs. “train” and “cow” vs. “sheep”, we propose to use the representation of new classes to enrich the variants of the distribution of old classes.

Firstly, the old object detector is utilized to construct a relation graph $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$ between old and new classes, where $\mathcal{V}$ are class nodes and $\mathcal{E}$ is the similarity between old and new classes. It is a confusion matrix on the data of new classes, which can be calculated as Eq. 2:

$$\begin{aligned} \varepsilon_{ij} &= \frac{\sum_{i=1}^{N_i} 1(M_{old}(x_i) = C_j^{Old})}{N_i}, \\ \Sigma_c &= \frac{(\sum_{i=1}^{C^{New}} \varepsilon_{i,c}\Sigma_i + \Sigma_c^{Old})}{2}, \end{aligned} \tag{2}$$

where 1 is the indicator function and ε_{ij} represents the ratio of the number of objects of the i^{th} new class predicted into the j^{th} old class. Σ_c^{Old} is calculated by the instance-level features in the detection head.

Then, we randomly sample augmented features F^a from the refined distribution as written in Eq. 3:

$$F^a \sim N(F, \Sigma_c), \tag{3}$$

where F is the instance-level features. The cross-entropy loss of the augmented features of old classes in the feature space of the detection head is defined as Eq. 4:

$$\mathcal{L}_{fea} = \frac{1}{\mathcal{C}_{old}} \sum_{k=1}^{\mathcal{C}_{old}} \frac{1}{M} \sum_{m=1}^{M} -\log\left(\frac{e^{w_k^T F^a_{k,m} + b_k}}{\sum_{c=1}^{\mathcal{C}_{all}} e^{w_c^T F^a_{c,m} + b_c}}\right). \tag{4}$$

M is the number of generated features. It is optimized with the upper bound derived from (Wang et al. 2021) when $M \to \infty$.

The total loss function of the whole framework is defined as:

$$\mathcal{L} = \mathcal{L}_{det} + \mathcal{L}_{fea} + \mathcal{L}_{dist}, \tag{5}$$

where $\mathcal{L}_{det}$ is the standard detection loss function in the object detector, $\mathcal{L}_{fea}$ and $\mathcal{L}_{dist}$ are the feature augmentation and distillation losses respectively. We use L1 loss for $\mathcal{L}_{dist}$.

Experiments

Experiment Setup

Datasets. The proposed method is evaluated on two benchmark datasets Pascal VOC 2007 and Microsoft COCO 2014. VOC2007 has 20 object classes, and we use the trainval subset for training and the test subset for evaluation. For COCO, the $80K$ images in the training set are used for training, and the minival (the first $5K$ images from the validation set) split is used for evaluation. There are two schemes to add new classes for evaluating our method: addition at once and sequential addition. In the following experiments, for fair comparisons with other methods, we crop the objects from the old training data without using any extra wild data.

Evaluation metrics. We use mean average precision (mAP) at 0.5 IoU threshold for VOC2007 and mAP across different IoU from 0.5 to 0.95 for COCO. The compared methods are fine-tuning and some recent related works (Chen, Yu, and Chen 2019; Hao et al. 2019; Shmelkov, Schmid, and Alahari 2017; Li et al. 2019; Zhang et al. 2020; Yang et al. 2022b; Joseph et al. 2021a,b). We list the results of these methods reported in their original papers, which are evaluated under the same settings as our proposed method without using any wild data. We also design “Baseline++” to evaluate different components, which uses L1 loss for feature distillation on instance-level features and logit distillation on output layers.

Implementation details. We use Faster R-CNN (Ren et al. 2015) with ResNet-50 (He et al. 2016) as the basic object detector. The old model is trained for 20 epochs, and the initial learning rate is set to 0.001 ($lr = 0.001$), and decays every 5 epochs with $gamma = 0.1$. The momentum is set to 0.9. The new model is trained for 10 epochs with $lr = 0.001$ and decays after 5 epochs. The confidence and IoU threshold for NMS are set to 0.5 and 0.3 respectively. The experiments are conducted on a single NVIDIA GeForce RTX 2080 Ti.

Components			mAP		
Copy-Paste	Object Aug.	Feature Aug.	19+1	15+5	10+10
			63.37	60.57	57.10
✓			68.91 (+5.54)	65.19 (+4.62)	60.60 (+3.50)
✓	✓		69.21 (+5.84)	66.55 (+5.98)	61.19 (+4.09)
✓	✓	✓	69.47 (+6.10)	66.62 (+6.05)	61.64 (+4.54)

Table 1: Ablation Study.

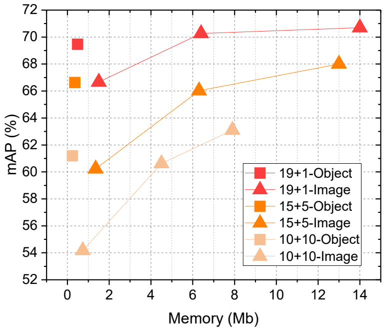

Figure 3: The precision of replaying images and objects on non-overlapped 19+1, 15+5 and 10+10 settings of VOC2007, respectively.

Method	19+1	15+5	10+10
Random	68.77	65.77	57.11
OSR	69.47 (+0.70)	66.62 (+0.85)	61.64 (+4.53)

Table 2: Comparison on object selection.

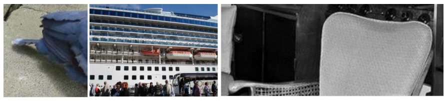

Figure 4: The randomly selected objects on 10+10 setting.

Ratio	0.1	0.2	0.5	0.7	1.0
mAP	69.04	69.3	69.47	69.54	69.43

Table 3: The ratio of the new synthetic samples.

Ablation Study

For the ablation study, the first 19, 15 and 10 classes are sorted in alphabetical order as old classes, and the remaining 1, 5 and 10 classes are corresponding new classes. To evaluate the performance on a non-co-occurrence setting, different from the commonly used settings in (Shmelkov, Schmid, and Alahari 2017), we select the images that only contain the objects of classes in this group, which means that the unlabeled objects of old classes do not appear in new data.

Table 1 lists the results of the variants of OSR for evaluating the effectiveness of different components. The first row is the results of our designed “Baseline++”. Compared with “Baseline++”, the copy-paste way to replay old objects can improve the performance by a large margin (4.55%) as shown in the second row, which will not increase the size of the training set. After adding the proposed object augmentation module, the mAP increases by 0.75% on average. As shown in the last row, the mAPs are also consistently improved on all settings when using feature augmentation, improving about 0.26% on average.

To verify the effectiveness of the proposed one-shot replay method, we compare it with replaying the original training images, which directly stores a subset of the original training data. For images, we randomly select 1, 5 and 10 images for each old class. For cropped objects, we only store one cropped object for each old class. Figure 3 presents the precision as the number of the stored samples increases. It can be seen that OSR achieves better mAP with negligible memory compared with replaying original images.

We also conduct experiments on the methods of object selection. As shown in Table 2, compared with random selection, the selection method in OSR can effectively improve the performance, especially on 10+10 setting. The reason may be that the representations learned on the first 10 classes are not robust, which may be easily interrupted by the randomly selected one-shot objects of old classes. We exemplify the selected occluded objects as shown in Figure 4, which may degrade the performance and change the learned decision boundary due to the lack of contextual information.

Table 3 presents the performance on different numbers of new synthetic samples. We calculate the number of new synthetic samples according to the total number of the original new training samples, where $N_{syn} = ratio \times N_{real}$. As can be seen, the best performance is achieved with $ratio = 0.7$. However, to save training time, we set the ratio to 0.5 for the trade-off between training time and accuracy.

In Figure 5, we verify the performance of feature augmentation module with or without analogy using the data of new classes (w New vs. w/o New). As can be seen, with the number of similar classes increasing, this module is more effective (larger improvements on the 10+10 setting).

Addition of Classes at Once

In this section, we evaluate the performance of adding new classes at once and compare the proposed method with the recent state-of-the-art incremental object detection methods. In this experiment, if the image contains the categories to be

Method	Model	19+1 Setting				15+5 Setting				10+10 Setting			
		Old	New	All	Mem.	Old	New	All	Mem.	Old	New	All	Mem.
Old	Faster R-CNN	73.4	-	73.4	-	74.3	-	74.3	-	88.0	-	88.0	-
Fine-tuning	Faster R-CNN	31.8	56.0	33.0	-	47.5	54.7	49.3	-	37.3	65.2	51.3	-
Shmelkov et al. 2017	Fast R-CNN	68.5	62.7	68.3	-	68.3	58.4	65.9	-	63.2	63.1	63.1	-
Chen et al. 2019	Faster R-CNN	68.3	60.0	67.9	-	-	-	-	-	65.0	60.0	62.5	-
Li et al. 2019	RetinaNet	66.3	40.4	65.0	-	-	-	-	-	67.5	68.4	67.9	-
Zhou et al. 2020	Faster R-CNN	70.5	53.0	69.6	-	-	-	-	-	63.5	60.0	61.8	-
Zhang et al. 2020	Faster R-CNN	65.6	64.0	65.5	-	-	-	-	-	65.8	61.7	63.8	-
Yang et al. 2022b	Faster R-CNN	70.5	59.6	70.0	-	69.6	59.2	67.0	-	66.3	66.0	66.2	-
Joseph et al. 2021a	Faster R-CNN	69.4	60.1	68.9	75	71.8	58.7	68.5	58	60.4	68.8	64.6	38
Joseph et al. 2021b	Faster R-CNN	70.9	57.6	70.2	14	71.7	55.9	67.8	13	68.4	64.3	66.3	7.9
OSR	Faster R-CNN	71.8	66.5	**71.5**	**0.5**	72.2	61.2	**69.5**	**0.4**	68.7	70.8	**69.8**	**0.3**
Upper(1-20)	Faster R-CNN	73.5	73.3	73.5	-	75.2	68.3	73.5	-	73.5	73.5	73.5	-

Table 4: Average precision (%) and memory usage (Mb) on the VOC2007 test dataset. Comparisons are conducted under different settings when 1, 5 or 10 classes are added at once.

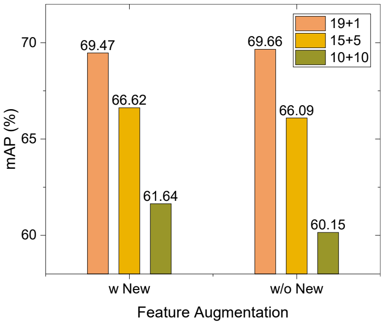

Figure 5: Comparison on feature augmentation with/without new classes for analogy.

detected, it will be selected for training or testing, so there is an overlap between the old and new data. As shown in Table 4, the per-class average precision on the VOC2007 test dataset is listed when 1, 5 and 10 new classes are added at once. The results of other methods are also listed, which are from their original papers.

On the 19+1 setting, Old(·) represents the old model trained on the data of the old classes. Here, except for the newly added layers for new classes, we initialize the parameters of the rest layers by the old model when performing incremental learning or fine-tuning. As can be seen, the performance of fine-tuning degrades a lot on old classes, which has caused severely catastrophic forgetting. OSR outperforms the competitive method by Joseph et al. (Joseph et al. 2021b) about 1.3% with only 0.49Mb memory. (Joseph et al. 2021a) and (Joseph et al. 2021b) store a balanced set of exemplars (50 and 10 complete images for each class respectively), taking about 75Mb and 14Mb on 19+1 setting. On the 15+5 setting, OSR also performs well compared with other methods. The mAP increases by 1.4% compared with results of (Joseph et al. 2021a), and memory takes up 0.36Mb. On the 10+10 setting, when adding more new classes, OSR still outperforms all methods and exceeds the second best method Li et al. (Li et al. 2019) about 1.9% and exceeds Joseph et al. about 3.5%.

Method	mAP@0.5	mAP@[0.5, 0.95]
Old(1-40)	54.4	32.5
Shmelkov et al. 2017	37.4	21.3
Zhou et al. 2020	36.8	22.7
Yang et al. 2022b	43.2	23.6
Joseph et al. 2021b	40.5	23.8
OSR	**45.2**	**25.4**
Upper(1-80)	50.1	29.8

Table 5: Average precision (%) on COCO minival dataset. Comparisons are conducted when 40 classes are added at once.

As can be seen, the performance of OSR is consistently improved compared with other methods on all settings. The mAPs after incremental learning are very close to the upper bound (Upper(1-20)), which is trained on the joint data of both old and new classes. It demonstrates that our proposed replay method can effectively mitigate catastrophic forgetting and reduce the memory usage for storing the samples of old classes.

To evaluate the performance of adding more classes, we conduct experiments on COCO, where the first 40 classes are the old classes and the remaining 40 classes are the new classes. The results are listed in Table 5. Compared with the distillation-based method (Shmelkov, Schmid, and Alahari 2017; Zhou et al. 2020; Joseph et al. 2021b), the simpler OSR can better mitigate catastrophic forgetting with only 0.86Mb extra memory.

Sequential Addition of Multiple Classes

In this experiment, we evaluate the performance of our method by adding classes sequentially for incremental learning. For the first setting, we also take 15 and 10 classes from VOC2007 sorted in alphabetical order as the old classes, and the remaining 5 and 10 classes are as new classes. Table 6

	Method	plant	sheep	sofa	train	tv
15+1+1+1+1+1	Replay Img	51.63	55.11	45.07	42.39	33.51
	OSR	70.54	67.99	64.96	63.75	63.13
	Method	table & dog	horse & mbike	person & plant	sheep & sofa	train & tv
10+2+2+2+2+2	Replay Img	47.81	41.40	20.77	21.81	21.95
	OSR	62.74	58.96	57.03	55.66	56.20

Table 6: Average precision (%) on VOC2007 test dataset when adding 5 or 10 new classes sequentially.

VOC2007	A	B	C	D	mAP
Baseline++	67.58	-	-	-	67.58
	48.63	73.67	-	-	61.15
	29.24	39.40	65.55	-	44.73
	21.91	24.86	37.66	45.35	32.45
OSR	67.58	-	-	-	67.58
	60.83	73.88	-	-	**67.35**
	48.13	51.35	66.61	-	**55.36**
	43.62	42.71	50.13	49.19	**46.41**
COCO	A	B	C	D	mAP
Baseline++	62.17	-	-	-	62.17
	50.45	22.18	-	-	36.31
	35.83	14.21	21.02	-	23.69
	23.17	10.69	15.67	26.23	18.94
OSR	62.17	-	-	-	62.17
	53.94	22.64	-	-	**38.29**
	49.50	18.63	22.31	-	**30.14**
	45.09	16.23	18.13	26.57	**26.50**

Table 7: Results on VOC2007 and COCO, when four groups are added sequentially.

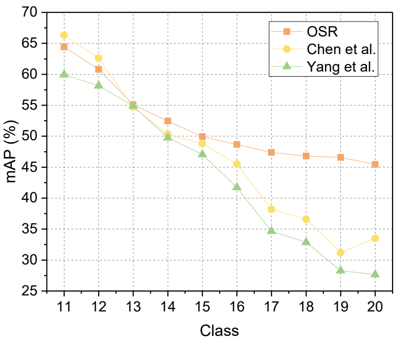

Figure 6: Comparison with the regularization-based method (Chen, Yu, and Chen 2019; Yang et al. 2022b) on the 10+1+...+1 setting of VOC2007.

lists the mAP(%) when adding 5 and 10 classes sequentially with 5 steps. We compare OSR with replaying one image for each old class (Replay Img), and the images are randomly selected to be stored with the original annotations. As can be seen, OSR achieves comparable performance with replay original images after multiple learning steps. The proposed method is also compared with the same Faster R-CNN based methods (Chen, Yu, and Chen 2019; Yang et al. 2022b) in ten-step learning, which are distillation-based methods. As shown in Figure 6, OSR can slow down the descent of the performance compared with the regularization-based methods with the increasing of the incremental learning steps, and it outperforms the results of Chen et al. (Chen, Yu, and Chen 2019) about 11.94% on the final learning step.

For the second setting, we split the training set of COCO into four groups: A, B, C and D. For each group, images that only contain the objects of classes in this group are selected. As is shown in Table 7, we compare OSR with Baseline++. OSR can achieve better performance than the method with only distillation on the non-overlapped scenario for sequential addition. It demonstrates that OSR can effectively compensate for the missing of old-class objects in the new training data, and this compensation can effectively mitigate catastrophic forgetting.

Discussion

The proposed one-shot replay is an augmentation method for IOD, which can be easily integrated with other regularization-based methods to further boost the performance. OSR can handle the problem of unavailable original training data and annotations. It can collect other easily obtained object-centric images in the wild by using the old detector to detect, requiring no manual bounding-box annotations for replay. In addition, the memory occupied by one object for each class is negligible, and the computational complexity is less than regularization-based methods. Therefore, OSR can be seen as an effective replay method to avoid complex computation and large memory size.

Conclusion

In this paper, rather than designing the complex regularization methods for preserving the learned knowledge, we propose a simple yet effective data replay method based on data and feature augmentation to improve the performance of incremental object detection. The easily accessible one cropped object for each old class is stored with smaller memory size. The proposed two augmentation modules can generate a diverse set of new samples and enrich the feature space, requiring no manual bounding-box annotations. Experimental results on VOC2007 and COCO demonstrate the effectiveness of the proposed method on incrementally learning to detect objects of new classes and mitigating catastrophic forgetting.

Acknowledgments

This work is supported by the Key Research Program of Frontier Sciences, CAS, Grant NO ZDBS-LY-7024, the Project of the public AI platform for science education supported by PKU-Baidu Fund, Grant NO 2020BD001, and the National Natural Science Foundation of China (62076195).

References

Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision*, 139–154.

Aljundi, R.; Chakravarty, P.; and Tuytelaars, T. 2017. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3366–3375.

Arbeláez, P.; Pont-Tuset, J.; Barron, J. T.; Marques, F.; and Malik, J. 2014. Multiscale combinatorial grouping. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 328–335.

Chen, L.; Yu, C.; and Chen, L. 2019. A new knowledge distillation for incremental object detection. In *2019 International Joint Conference on Neural Networks*, 1–7. IEEE.

Delange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; and Tuytelaars, T. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Dwibedi, D.; Misra, I.; and Hebert, M. 2017. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 1301–1310.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2): 303–338.

Fang, H.-S.; Sun, J.; Wang, R.; Gou, M.; Li, Y.-L.; and Lu, C. 2019. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 682–691.

French, R. M. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4): 128–135.

Ghiasi, G.; Cui, Y.; Srinivas, A.; Qian, R.; Lin, T.-Y.; Cubuk, E. D.; Le, Q. V.; and Zoph, B. 2021. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2918–2928.

Girshick, R. 2015. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, 1440–1448.

Goodfellow, I. J.; Mirza, M.; Xiao, D.; Courville, A.; and Bengio, Y. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

Hao, Y.; Fu, Y.; and Jiang, Y.-G. 2019. Take Goods from Shelves: A dataset for class-incremental object detection. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, 271–278.

Hao, Y.; Fu, Y.; Jiang, Y.-G.; and Tian, Q. 2019. An end-to-end architecture for class-incremental object detection with knowledge distillation. In *2019 IEEE International Conference on Multimedia and Expo*, 1–6. IEEE.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 770–778.

Joseph, K.; Khan, S.; Khan, F. S.; and Balasubramanian, V. N. 2021a. Towards open world object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5830–5840.

Joseph, K.; Rajasegaran, J.; Khan, S.; Khan, F. S.; and N Balasubramanian, V. 2021b. Incremental Object Detection via Meta-Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526.

Kisantal, M.; Wojna, Z.; Murawski, J.; Naruniec, J.; Cho, K.; et al. 2019. Augmentation for small object detection. In *In 9th International Conference on Advances in Computing and Information Technology*, volume 9.

Li, D.; Tasci, S.; Ghosh, S.; Zhu, J.; Zhang, J.; and Heck, L. 2019. RILOD: Near real-time incremental learning for object detection at the edge. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, 113–126.

Li, Z.; and Hoiem, D. 2017. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12): 2935–2947.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, 740–755. Springer.

Liu, Y.; Hong, X.; Tao, X.; Dong, S.; Shi, J.; and Gong, Y. 2022. Model Behavior Preserving for Class-Incremental Learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Mai, Z.; Li, R.; Jeong, J.; Quispe, D.; Kim, H.; and Sanner, S. 2021. Online continual learning in image classification: An empirical survey. *arXiv preprint arXiv:2101.10423*.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, 109–165. Elsevier.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2001–2010.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 91–99.

Shin, H.; Lee, J. K.; Kim, J.; and Kim, J. 2017. Continual learning with deep generative replay. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2994–3003.

Shmelkov, K.; Schmid, C.; and Alahari, K. 2017. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE International Conference on Computer Vision*, 3400–3409.

Wang, X.; Huang, T.; Gonzalez, J.; Darrell, T.; and Yu, F. 2020. Frustratingly Simple Few-Shot Object Detection. In *International Conference on Machine Learning*, 9919–9928. PMLR.

Wang, Y.; Huang, G.; Song, S.; Pan, X.; Xia, Y.; and Wu, C. 2021. Regularizing deep networks with semantic data augmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wang, Y.; Huang, Z.; and Hong, X. 2022. S-Prompts Learning with Pre-trained Transformers: An Occam's Razor for Domain Incremental Learning. In *Advances in Neural Information Processing Systems*.

Xu, Z.; Meng, A.; Shi, Z.; Yang, W.; Chen, Z.; and Huang, L. 2021. Continuous Copy-Paste for One-Stage Multi-Object Tracking and Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15323–15332.

Yang, D.; Zhou, Y.; Shi, W.; Wu, D.; and Wang, W. 2022a. RD-IOD: Two-Level Residual-Distillation-Based Triple-Network for Incremental Object Detection. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 18(1): 1–23.

Yang, D.; Zhou, Y.; Zhang, A.; Sun, X.; Wu, D.; Wang, W.; and Ye, Q. 2022b. Multi-view correlation distillation for incremental object detection. *Pattern Recognition*, 131: 108863.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

Zhang, J.; Zhang, J.; Ghosh, S.; Li, D.; Tasci, S.; Heck, L.; Zhang, H.; and Kuo, C.-C. J. 2020. Class-incremental learning via deep model consolidation. In *The IEEE Winter Conference on Applications of Computer Vision*, 1131–1140.

Zhou, W.; Chang, S.; Sosa, N.; Hamann, H.; and Cox, D. 2020. Lifelong Object Detection. *arXiv preprint arXiv:2009.01129*.

Zhu, F.; Cheng, Z.; Zhang, X.-Y.; and Liu, C.-l. 2021. Class-Incremental Learning via Dual Augmentation. *Advances in Neural Information Processing Systems*, 34: 14306–14318.

Zitnick, C. L.; and Dollár, P. 2014. Edge boxes: Locating object proposals from edges. In *Proceedings of the European Conference on Computer Vision*, 391–405. Springer.