RD-IOD: Two-Level Residual-Distillation-Based Triple-Network for Incremental Object Detection

DONGBAO YANG, Chinese Academy of Sciences and University of Chinese Academy of Sciences
YU ZHOU, Chinese Academy of Sciences
WEI SHI, Carleton University
DAYAN WU and WEIPING WANG, Chinese Academy of Sciences

As a basic component in multimedia applications, object detectors are generally trained on a fixed set of classes that are pre-defined. However, new object classes often emerge after the models are trained in practice. Modern object detectors based on Convolutional Neural Networks (CNN) suffer from catastrophic forgetting when fine-tuning on new classes without the original training data. Therefore, it is critical to improve the incremental learning capability on object detection. In this article, we propose a novel Residual-Distillation-based Incremental learning method on Object Detection (RD-IOD). Our approach rests on the creation of a triple-network based on Faster R-CNN. To enable continuous learning from new classes, we use the original model as well as a residual model to guide the learning of the incremental model on new classes while maintaining the previous learned knowledge. To better maintain the discrimination between the features of old and new classes, the residual model is jointly trained with the incremental model on new classes in the incremental learning procedure. In addition, a two-level distillation scheme is designed to guide the training process, which consists of (1) a general distillation for imitating the original model in feature space along with a residual distillation on the features in both image level and instance level, and (2) a joint classification distillation on the output layers. To well preserve the learned knowledge, we design a 2-threshold training strategy to guide the learning of a Region Proposal Network and a detection head. Extensive experiments conducted on VOC2007 and COCO demonstrate that the proposed method can effectively learn to incrementally detect objects of new classes, and the problem of catastrophic forgetting is mitigated. Our code is available at https://github.com/yangdb/RD-IOD.

CCS Concepts: • **Computing methodologies → Object detection**; **Transfer learning**; **Lifelong machine learning;**

Additional Key Words and Phrases: Object detection, incremental learning, residual distillation

This work was supported by the Open Research Project of the State Key Laboratory of Media Convergence and Communication, Communication University of China, China (No. SKLMCC2020KF004), the Beijing Municipal Science & Technology Commission (Z191100007119002), the Key Research Program of Frontier Sciences, CAS, Grant No. ZDBS-LY-7024, and the National Natural Science Foundation of China (No. 62006221).

Authors' addresses: D. Yang, Institute of Information Engineering, Chinese Academy of Sciences, A No. 89, Minzhuang Road, Haidian District, Beijing, China, 100093 and School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China, 100049; email: yangdongbao@iie.ac.cn; Y. Zhou (corresponding author), D. Wu, and W. Wang, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China; emails: {zhouyu, wudayan, wangweiping}@iie.ac.cn; W. Shi, School of Information Technology, Carleton University, 1125 Colonel By Drive, Ottawa, Canada; email: wei.shi@carleton.ca.
Updated author affiliation: Dongbao Yang, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China and School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2022 Association for Computing Machinery.

1551-6857/2022/01-ART18 $15.00
https://doi.org/10.1145/3472393

ACM Reference format:
Dongbao Yang, Yu Zhou, Wei Shi, Dayan Wu, and Weiping Wang. 2022. RD-IOD: Two-Level Residual-Distillation-Based Triple-Network for Incremental Object Detection. *ACM Trans. Multimedia Comput. Commun. Appl.* 18, 1, Article 18 (January 2022), 23 pages.
https://doi.org/10.1145/3472393

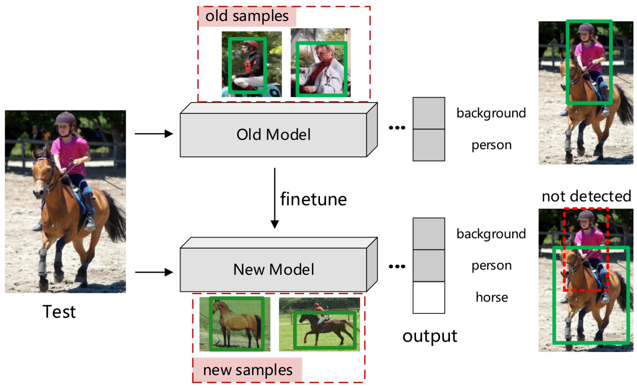
1 INTRODUCTION

Object detection is a basic computer vision task that is the foundation of many multimedia applications, such as autonomous driving, text spotting, image captioning, object tracking, and video classification [7, 27, 29, 34–36, 43, 44, 48, 49]. In real applications, new object classes often emerge after the object detection model has been trained on a prepared dataset with fixed classes. Due to the storage burden, the privacy concern and the limited availability of the old data, as well as the time consumption, sometimes it is impractical to train the model from scratch with both old and new data. Despite the state-of-the-art results achieved, modern object detectors based on **Convolutional Neural Networks (CNNs)** suffer from *catastrophic forgetting* [9, 11, 30] in learning new object classes without the original data. It is therefore necessary to improve the ability of incremental learning for object detectors.

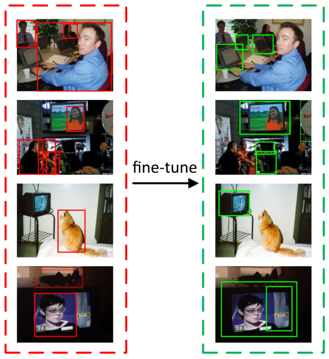
Fine-tuning is a common way to adapt the original model to new classes. As shown in Figure 1(a), it is achieved by replacing the output layer with new classes or by adding units in the output layer specifically for new classes. Unfortunately, if the model is fine-tuned only on the samples of new classes, the model will quickly forget the old classes. In Figure 1(b), we present detection results of the original model trained on old classes (cat, person, table, etc.) versus the results of fine-tuning on one new class (tvmonitor). It is clear that the performance of the model after fine-tuning is severely degraded due to the absence of old data. Intuitively, training the model with both old and new data should solve this problem. However, it will likely increase the execution time as well as the storage burden for storing old data. In particular, the training dataset for a pre-trained model is not always available for a new task due to data privacy concerns. Therefore, it is necessary to develop incremental learning methods for object detection that enables continuous learning from new data while preserving the previously learned knowledge.

Many studies on incremental learning mainly focus on image classification. Based on different optimization techniques used to overcome catastrophic forgetting, incremental learning methods can be divided into two categories [16]: (1) preserving significant parameters of the original model [1, 20, 45] and (2) preserving the knowledge of the original model through knowledge distillation [2, 18, 24, 37, 38]. Due to the complexity of designing a metric to evaluate the importance of all parameters in an original model, we continue the exploration on the second direction for object detection tasks. More precisely, this approach utilizes the knowledge learned from the original model to guide the training of the new model by calculating distillation losses.

Contrary to image classification, object detection involves distinguishing foreground from complex background, and performing the precise localization of objects. These tasks increase the difficulty of incremental learning. Previous incremental object detection methods [6, 12, 13, 23, 40, 46] mainly optimize directly the incremental model to imitate the original model in some important parts such as the backbone and the output layers to preserve the knowledge learned from the old data. This is achieved by constraining the features and outputs between these two models to be similar. However, the image-level features of object detectors contain the information of both foreground and background. The activations with high responses in the image-level features of the original model are related to the old classes. Therefore, the distillation on image-level features should aim to preserve the high responses of old classes and add responses for new



(a) Illustration of fine-tuning



(b) Detection results

Fig. 1. Fine-tuning on new classes. (a) The original model is trained on an old class (person), and the new model is fine-tuned from the original model on a new class (horse) by adding one unit on the output layer. The person in the test image cannot be detected after fine-tuning. (b) Detection results: The left column shows the detection results of the model trained on old classes (cat, person, table, etc.), and the right column shows the detection results of the model fine-tuned only on a new class (tvmonitor).

classes simultaneously. Simply imitating the original model will suppress the capability of the incremental model on learning the discriminative features from both old and new classes. In fact, due to the difference between the original model and the incremental model on the categories to be detected, the responses on the feature maps are different. More precisely, the obvious residual errors should be maintained on the features of the original model as well as the incremental model rather than preserving only the similarities between them. To maintain the residual errors in the incremental learning procedure, a feasible way is to use an assistant model (also a kind of residual model) to explicitly model the difference between the original model and the incremental model. Because the essential differences between these two models are on the new information emerged from the new classes, we believe that minimizing the loss for detecting objects of new classes in the residual model should further improve the optimization result.

In this article, we propose a novel triple-network-based incremental object detector that is based on Faster R-CNN [39]. Contrary to designing complex model structure with object-detector-specific components to handle new classes, the proposed method mainly focuses on designing the distillation method for better incremental learning of new classes while maintaining the previously learned knowledge when the old dataset is not available. It is a simple yet effective method that is designed considering the common characteristics of object detection, which can be generalized to a wide range of object detection frameworks. Three detection models work jointly to adapt the original network trained on old classes to new classes, and to ensure that the performance on the old classes does not degrade without using the old training data. To train an incremental model that is responsible for detecting both old and new classes, a frozen copy of the original Faster R-CNN trained on old classes is utilized to provide the knowledge of old classes including feature distributions, output logits, and pseudo ground-truth. In addition, a novel residual model is proposed to assist in the incremental learning procedure. To preserve the previous learned knowledge, we design a novel distillation scheme, which includes a new general feature distillation loss, a residual distillation mechanism, and a joint classification distillation loss applied in the feature space and on the output layers separately. The contributions of this article are summarized as follows:

- We propose a novel triple-network-based incremental learning method for object detection, which explicitly models the inherent residual information between the features of the incremental model and the original model by jointly training a residual model. The original model provides the knowledge learned on the data of old classes, and a residual model is jointly trained to obtain the knowledge from the data of new classes and maintain the residual information in the incremental learning procedure.
- To maintain the performance on old classes and improve the scaling ability of the detection model on new classes, a novel distillation scheme is elaborately designed in both image level and instance level. A two-level residual distillation is proposed to guide the learning of the triple-network in feature space with a new method to calculate the difference between image-level features. Furthermore, a joint classification distillation loss is utilized to preserve the learned knowledge from both old and new classes in the output layers. In addition, a 2-threshold training strategy is proposed to better utilize the knowledge obtained from the original model.
- Extensive experiments are conducted on VOC2007 [8] and COCO [26], and the results demonstrate that the proposed approach is effective for incremental object detection and achieves promising results compared with previous methods.

2 RELATED WORK

Incremental learning is a significant problem in machine learning [5, 22, 31, 33]. Meanwhile, the capability to continuously learn from newly emerging objects is also essential for many practical multimedia applications. Recently, with the success of deep learning, many researchers have paid more attention to improve incremental learning techniques in deep neural networks.

Incremental classification. Most existing incremental learning methods for vision tasks mainly focus on image classification, which means to continuously update the image classifier to recognize new classes without decreasing the accuracy on old classes. Existing works can be divided into two categories based on the optimization directions in preserving learned knowledge [16]: parameter based and distillation based.

Some works are based on preserving important parameters of the network to maintain the performance on old tasks or classes. Jung et al. [17] present a method to maintain the performance on old task by freezing the weights of the softmax layer and minimizing the distance between the features extracted from target and source networks, respectively. The limitation of this approach is that the weights for new and old tasks may be in conflict, and the parameters cannot be updated, which results in degraded performance in learning new tasks. EWC [20] is a prominent work in this category, which sets the weights of the new model by those of the original model according to the importance of weights, and the approach remembers the old tasks by selectively slowing down learning on the weights that are important for those tasks. MAS [1] accumulates an importance measure for each parameters of the network based on the sensitivity of the predicted output function to the changes in parameters. The changes of important parameters are penalized when learning a new task. Zenke et al. [45] introduce intelligent synapses to accumulate task relevant information over time and exploit this information to rapidly store new memories without forgetting the old ones. The limitation of these works is that it is hard to design a metric to evaluate the importance of all parameters.

The distillation-based method is the other representative category of incremental learning, where knowledge distillation is used to transfer knowledge from the original network to the new network. Knowledge distillation is defined to utilize the supervised information provided by a teacher model to guide the training of a student model to mimic the teacher model by controlling the distillation loss. LwF [24] is the first method to use knowledge distillation for incremental learning, which utilizes a modified cross-entropy loss to preserve original knowledge with only examples from the new task. iCaRL [38] proposes to jointly learn feature representation and classifier by combining representation learning and knowledge distillation, and a small set of exemplars is selected to perform nearest-mean-of-exemplars classification. Rannen et al. [37] propose an auto-encoder-based method to retain the knowledge from old tasks, which prevents the reconstructions of the features from changing and leaves space for the features to adjust. Sun et al. [41, 42] propose to maintain a lifelong dictionary, which is used to transfer knowledge to learn each new metric learning task.

Transfer learning. Transfer learning, which is also related to incremental object detection, uses the knowledge acquired from one task to help the training of other tasks. Fine-tuning is a representative paradigm of transfer learning, which is frequently used in the initialization of the backbone in object detection models with a CNN trained on ImageNet [21]. An alternative to transfer knowledge from one network to another is distillation. Hinton et al. [15] transfer the knowledge from a large network to a small network using distillation, which encourages the responses of these two networks to be similar. However, transfer learning needs data for both old and new tasks to maintain the performance on the old tasks; otherwise, the performance will degrade severely when old data is not available.

Knowledge distillation is a simple and widely used technique that is initially proposed to perform model compression, where the student model is lighter than the teacher model. Knowledge distillation utilizes the knowledge acquired in the teacher model as supervisory signals to train the student model using the same labeled data. Thus, it can effectively improve the performance of the student model. Differently, incremental learning methods use knowledge distillation to alleviate performance degradation on the old classes (catastrophic forgetting) where the old data is no longer available when new object classes emerge.

Incremental object detection. Shmelkov et al. [40] propose the first incremental object detector based on Fast R-CNN [10] by applying knowledge distillation without using previous training data. First, it uses EdgeBoxes [51] and MCG [3] to pre-compute proposals. Then, these proposals are fed into the detection head to predict their categories. The model is trained with distillation losses applied onto the outputs of the final classification and regression layers to preserve the model's ability to detect old classes. The reason they use Fast R-CNN rather than Faster R-CNN is that they need class-agnostic proposals, but the **Region Proposal Network (RPN)** is class sensitive. However, this method is not end-to-end and the proposal generation procedure is not learnable and not real time, which prevents it from being adapted for use with most modern object detectors.

Recently, several end-to-end incremental object detection methods [6, 12, 13, 23] have been proposed based on Faster R-CNN [39]. Chen et al. [6] propose a hint loss to minimize the difference between the feature maps of the old and the incremental models. Furthermore, a confidence loss is used to suppress the generation of low confidence bounding boxes. Hao et al. [12] introduce a hierarchical large-scale retail object detection dataset called *TGFS* and present a class-incremental object detector that utilizes an exemplar set with a fixed amount of old data for training. Hao et al. [13] use a frozen duplication of the RPN to preserve the knowledge gained from the old classes, and a feature-changing loss is proposed to reduce the difference between the feature maps of the old and new classes. Li et al. [23] extract three types of knowledge from the original model to mimic its behaviors on object classification, bounding box regression, and feature extraction. Zhang et al. [46] propose a dual distillation training function that pre-trains a separate model only for the new classes such that a student model can learn from two teacher models simultaneously. In addition, a novel work on an incremental few-shot object detector based on CentreNet [50] is proposed by Perez-Rua et al. [32]. The few-shot setting is more challenging than the many-shot setting in incremental object detection, and the problem of incremental object detection on the many-shot setting has not been well resolved. In addition, the original structure of CentreNet is redesigned for few-shot learning. In our work, we mainly focus on general incremental learning for object detection, when only an original model trained on the old classes and data of new classes are available.

Incremental object detection and few-shot object detection aim to solve different problems. Incremental learning mainly focuses on the stability-plasticity dilemma, which means the network is required to have the stability to retain its performance on old classes and plasticity to learn new knowledge from new classes. In contrast to this, few-shot object detection aims to solve the problem where only a few annotated examples are available for each object category to train a detection model. Recent few-shot object detection methods [52] are mainly based on meta-learning or fine-tuning to adapt a model trained on abundant base classes to learn to detect new classes with few annotated examples. However, these methods cannot detect old classes if the data of old classes is not available.

The common approach of the preceding incremental object detection methods for preserving the learned knowledge is to imitate the important activations of the original model. In our proposed method, we not only let the incremental model imitate the important activations of the original model but also introduce a residual model trained simultaneously on the new classes. The latter is designed to (1) model explicitly the inherent differences between the original model and the incremental model, and (2) maintain the discrimination of the features of both old and new classes.

3 TRIPLE-NETWORK-BASED INCREMENTAL DETECTOR

3.1 Overview

In this article, we propose an end-to-end incremental object detector to continuously learn from new data without having access to the old data. Figure 2 illustrates the entire training framework of the proposed method. The proposed triple-network includes three detection networks. **Original Model (OriM)** is a frozen copy of the original model trained on the old data. This model provides the knowledge of the old classes, including the activations on feature maps, detection results, and distributions of the output layers. **Incremental Model (IncM)** is adapted to detect both old and new classes with the annotations of new data as well as the learned knowledge from OriM. The detection results from OriM, which are regarded as the pseudo ground-truth, are combined with the annotations of new data to update IncM. **Residual Model (ResM)** is an assistant model jointly trained with IncM to detect new classes. To better preserve the knowledge of old classes and maintain the discrimination between the features of old and new classes, a new distillation scheme is designed to add constraints onto the incremental learning procedure of IncM in both the image level and instance level. In the image level, a general feature distillation loss and a residual distillation loss on the features of the backbones are computed and used. In the instance level, we propose a residual distillation loss for the *pooled features* in the detection heads and a joint classification distillation loss for the output layers. The proposed incremental object detector is described in detail in the following section.

The triple-network that we propose is based on Faster R-CNN, which is an end-to-end proposal-based object detector. In the incremental learning stage, the parameters of IncM are initialized from OriM excluding the weights and bias of newly added output layers for the new classes, which are initialized randomly. We feed the new data into the triple-network, then OriM generates bounding boxes that are filtered using a confidence threshold and per-category **Non-Maxima Suppression (NMS)**. The remaining bounding boxes after the first filter that have

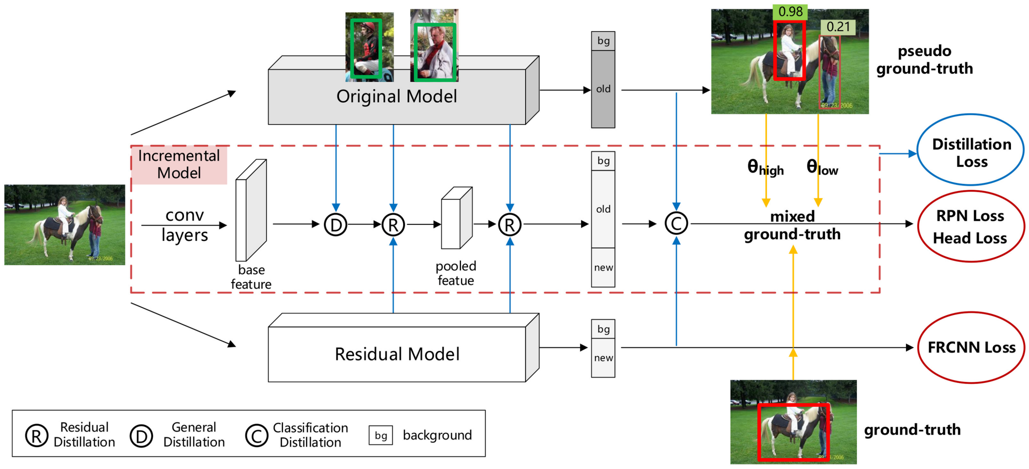


Fig. 2. The framework of the proposed end-to-end incremental object detection method. There are three detection models: (1) *Original Model* is a frozen copy of the originally trained model, which is used to generate the pseudo ground-truth of the old classes and supervised information for the training data of the new classes; (2) *Incremental Model* is a model that is fine-tuned after incremental learning from new classes while preserving the previous learned knowledge by controlling the distillation losses; and (3) *Residual Model* is used to learn the residual between the original model and the incremental model, as well as to learn to detect new classes.

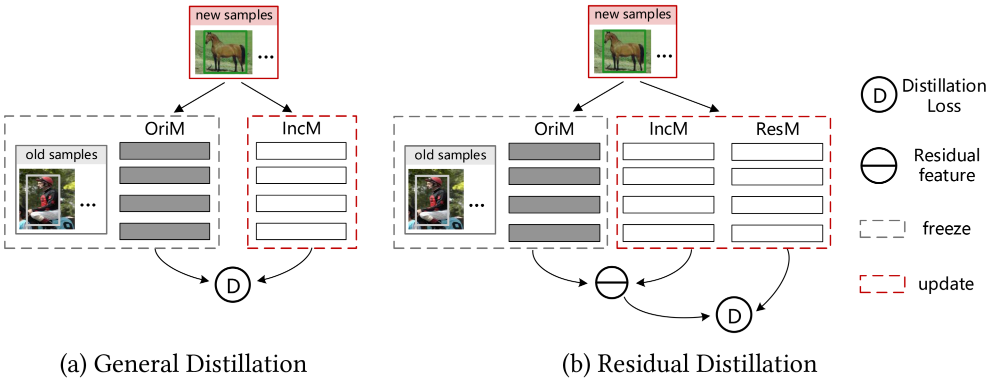
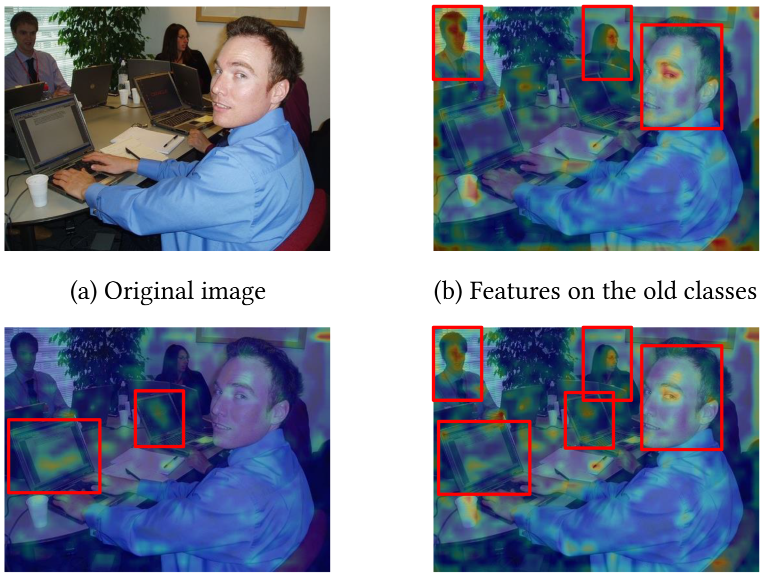


Fig. 3. Illustration of general distillation (a) and residual distillation (b).

high **Intersection-over-Unions (IoUs)** with any ground-truth from the new data are further filtered. Finally, the remaining bounding boxes are used as the pseudo ground truth to train IncM together with the original annotations of the new data.

To preserve the learned knowledge, many methods generally train IncM to mimic the feature and outputs of OriM. As shown in Figure 3(a), the general distillation calculates the difference between OriM and IncM, and minimizes the difference simultaneously. This constraint will restrict the capability of IncM to learn the unique characteristics for old and new classes, respectively. The activation-based spatial attention maps are visualized by $F_{att} = \sum_{i=1}^{C} |F_i|$ as shown in Figure 4, where F is the feature from the backbone and C is the number of channels. As can be seen, the



(a) Original image (b) Features on the old classes

(c) Features on the new classes (d) Merged features

Fig. 4. Visualization of the features from the backbones of Faster R-CNN trained on an old class (person) (a) and a new class (tvmonitor) (c), respectively. (d) The features after merging (b) and (c).

high responses on the feature maps of the detection models are related to the classes the models have been trained on. As shown in Figure 4(b) and Figure 4(c), the models are trained on old and new classes with the high responses received on an old class (person) and a new class (tvmonitor), respectively. In our incremental learning procedure, we hope the features of IncM have high responses on both old and new classes in images with the merged features shown in Figure 4(d). In other words, a residual information should be maintained between the features of IncM and OriM, which can be seen as a kind of information about new classes. Therefore, to train IncM in an end-to-end way during the incremental learning stage, we train ResM jointly with IncM on new classes, aiming to maintain the differences between the old and new classes. Meanwhile, ResM should mimic the residual features between OriM and IncM, which can be seen as the representation of the new classes. The illustration of our proposed residual distillation is shown in Figure 3(b). The backbone of ResM is initialized by a pre-trained ResNet on ImageNet, and the remaining parameters are initialized randomly.

In our proposed method, OriM and ResM are used together to assist in the training of IncM, which constitutes an end-to-end incremental learning method. In the inference procedure, only IncM is used to get the detection results. Therefore, our method will not change the original structure of the detection model, and it will not increase the execution time nor the space complexity during inference.

3.2 Image-Level Distillation

The distillation on image-level features aims to preserve the high responses of old classes and add the responses of new classes simultaneously. To achieve this goal, we propose to use general distillation and residual distillation on the image-level feature space of the backbones simultaneously, as shown in Figure 5.

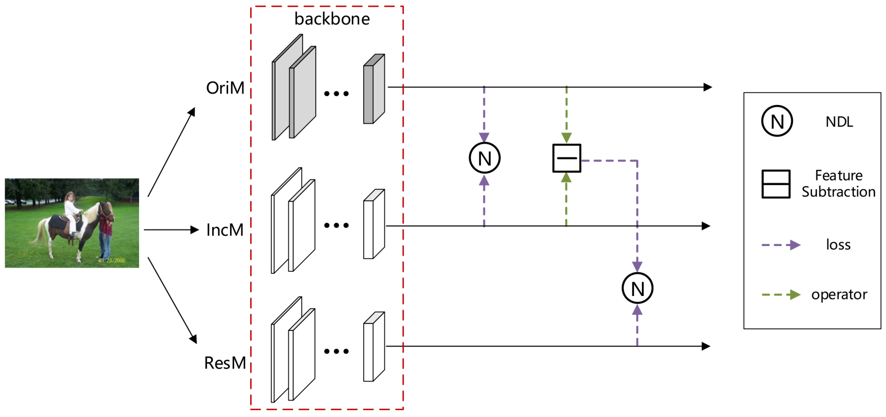


Fig. 5. Detailed illustration of image-level distillation. NDL denotes the proposed L2-NDL. Dark gray and white indicate the frozen and updated parameters, respectively.

The general distillation is used to imitate the behavior of OriM, and the residual distillation is used to maintain the discrimination between the features of IncM and OriM. The general distillation on the image-level features aims to preserve more knowledge of old classes and backgrounds learned from the old data. It is used to minimize the difference between the original and incremental models. However, in the incremental learning process, a certain amount of difference should be maintained by using the proposed residual distillation. This is because there is a trade-off between preserving the learned knowledge and learning new knowledge from the new classes, which must be balanced.

Intuitively, freezing the parameters of the backbone of IncM, and only updating the parameters of classifier and regressor for the new classes, is the best way to preserve the knowledge of the old classes. However, the performance of IncM will be degraded because the ability of the detector to learn new classes is restricted. On the contrary, if we directly fine-tune the backbone on new data, the parameters will update toward the new classes and deviate from the old classes.

To preserve the learned knowledge of old classes, distillation is often used to maintain feature similarity between OriM and IncM. It is achieved by forcing the activations of the backbones between IncM and OriM to be similar. Due to the fact that the features of backbones are correlated to both foreground and background, which are different for the old and the new classes, some discriminative responses of the new classes also should be maintained in the incremental learning procedure. Therefore, we design a new way to calculate the differences between the features of backbones, which is an L2-**Norm-based Distillation Loss (NDL)** as defined in Equation (2):

$$M(F)(i,j) = \frac{1}{c}\sum_{k=1}^{c} F(k,i,j), \quad F \in R^{c\times h\times w}, \quad M \in R^{h\times w}, \tag{1}$$

$$NDL(F_a, F_b) = |\|M(F_b)\|_2 - \|M(F_a)\|_2|. \tag{2}$$

$F \in R^{c\times h\times w}$ represents the feature map. c represents the number of channels, h and w represent the spatial dimensions, and M is the mean of F along channels, where (i,j) represents a coordinate on the feature map. L1-loss ($|\cdot|$) is used to penalize the differences between the L2-norm ($\|\cdot\|_2$) of $M(F_a)$ and $M(F_b)$. Compared with the L1 distance directly computed between the two feature maps, this method considers the original 2D information on each feature map as a whole, not just pointwise. The distillation loss between the backbones of OriM and IncM is calculated as Equation (3):

$$D_{fea}(F_{IncM}, F_{OriM}) = NDL(F_{IncM}, F_{OriM}). \tag{3}$$

To maintain IncM's capability to preserve the knowledge learned from the old classes while learning the new classes simultaneously, we propose a residual distillation method (D_{res}). It can be divided into two parts D_{res}^{base} and D_{res}^{pool}, which represents the residual losses applied on the feature maps of the backbone and detection head, respectively.

For the feature maps of the backbones, the residual between OriM and IncM is expressed as Equation (4):

$$F_{res} = F_{IncM} - F_{OriM}. \tag{4}$$

The difference is then calculated between the residual F_{res} and the feature map of ResM F_{ResM}, as shown in Equation (5):

$$D_{res}^{base}(F_{res}, F_{ResM}) = NDL(F_{res}, F_{ResM}). \tag{5}$$

D_{fea} and D_{res} are designed for two purposes: (1) preserve the learned knowledge from the data of old classes, and (2) maintain some difference between the features of IncM and OriM. IncM is expected to strike a balance between these two purposes.

3.3 Instance-Level Distillation

The distillation on the instance-level features aims to maintain the discrimination of the instances of different classes and add responses for the patterns of new classes. Therefore, at the instance level, the distillation losses are applied in two places in the network: (1) the pooled feature space and (2) the output layers of the detection head. Figure 6 gives a detailed illustration of the application of instance-level distillation losses. To apply the residual mechanism on the pooled features in the detection head, we design a new residual distillation loss for the instance-level features. **Regions-of-Interests (RoIs)**, generated from IncM, are provided to the detection heads of OriM, IncM, and ResM simultaneously. After RoI-pooling and several convolution layers, we obtain the final pooled features for the proposals of OriM, IncM, and ResM separately.

Differently from the feature distillation loss in the backbone, L1-loss is directly applied for computing the distance between feature maps at the instance level. Due to the fact that the background and all instances in an image are regarded as a whole in the backbone, our proposed NDL distillation method (Equation (2)) is more appropriate for the image-level features. On the contrary, L1-loss is more suited to measuring the distance between the features of the same instance.

$$\begin{gathered} P_{res} = P_{IncM} - P_{OriM} \\ D_{res}^{pool}(P_{res}, P_{ResM}) = |P_{res} - P_{ResM}|, \end{gathered} \tag{6}$$

where P represents the pooled features.

For the output layers of the detection head, the weights and biases of the classification layers can be considered as the high-level semantic representation of the classes. The parameters of classification layers corresponding to the old classes are initialized in OriM and used to maintain the representation. In the training of IncM, the classifier is fine-tuned for learning new classes, and the classification outputs from IncM are constrained to generate similar distributions to OriM and ResM for old and new classes, respectively. This is done to ensure the preservation of the learned knowledge. We then calculate according to the following equations: (1) the L2-loss between the softmax outputs of the old classes and the background from the classification layers of OriM and IncM, and (2) the softmax outputs of the new classes from the classification layers of

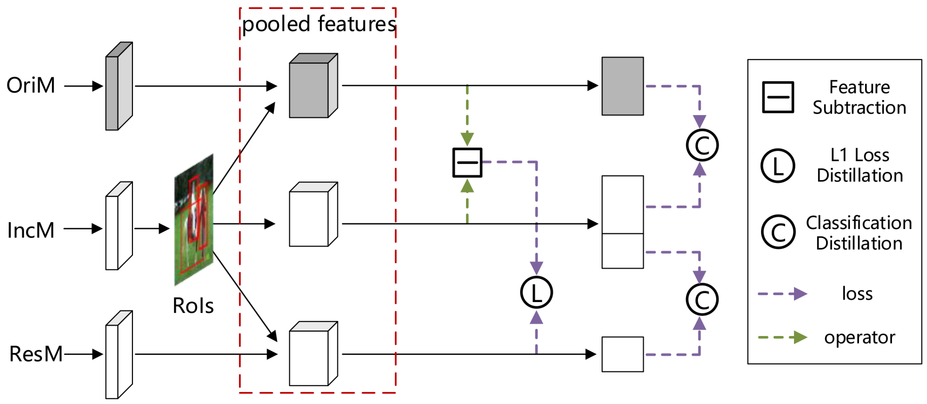


Fig. 6. Detailed illustration of instance-level distillation. Dark gray and white indicate the frozen and updated parameters, respectively.

ResM and IncM.

$$\begin{aligned} C_{IncM}^{old,i} &= \frac{exp(y^i)}{\sum_{j=0}^{N_{old}} exp(y^j)}, C_{OriM}^{i} = \frac{exp(p_{OriM}^i)}{\sum_{j=0}^{N_{old}} exp(p_{OriM}^j)} \\ C_{IncM}^{new,i} &= \frac{exp(y^i)}{\sum_{j=N_{old}+1}^{N_{old}+N_{new}+1} exp(y^j)}, C_{ResM}^{i} = \frac{exp(p_{ResM}^i)}{\sum_{j=1}^{N_{new}} exp(p_{ResM}^j)}, \end{aligned} \tag{7}$$

$$D_{cls} = \frac{1}{N_{old}+1} \sum_{k=0}^{N_{old}} \left(C_{IncM}^k - C_{OriM}^k\right)^2 + \frac{1}{N_{new}} \sum_{k=1}^{N_{new}} \left(C_{IncM}^{k+N_{old}} - C_{ResM}^k\right)^2, \tag{8}$$

where $C_{OriM} \in R^{N_{old}+1}$, $C_{ResM} \in R^{N_{new}+1}$, and $C_{IncM} \in R^{N_{old}+N_{new}+1}$ are the classification outputs of OriM, ResM, and IncM, respectively. N_{old} and N_{new} are the numbers of old and new classes, respectively.

3.4 2-Threshold Training

For the end-to-end two-stage object detectors, it is difficult to preserve the performance on the old classes when directly adapting OriM to the new classes without using the old data due to the usage of the class-sensitive RPN and detection head. Intuitively, the detection results of OriM on the new training data may provide useful knowledge acquired from training on the old classes to the incremental learning. Therefore, we utilize the detection results from the OriM as the pseudo ground-truth to keep the ability to detect old classes when performing training on new data with new classes. Most importantly, the confidence threshold is an important hyper-parameter to set up to obtain the pseudo ground-truth, which has a great influence on the performance of IncM. A high threshold may discard some potential object-like proposals identified from learning on the old data, and a low threshold may contain many false positives, which confuse the classifier of IncM and eventually lead to a detector with degraded performance.

It is well known that in two-stage object detection methods, the RPN generates region proposals from complex backgrounds with high recall, and the detection head is supposed to accurately classify and regress these proposals. To better preserve the knowledge learned from the old data, in the proposed incremental learning, we train these parts of IncM for different purposes: (1) the RPN is trained to preserve its performance on OriM for generating more object-like proposals on

ALGORITHM 1: 2-Threshold Training Strategy

```
Require: The incremental model $IncM$, original model $OriM$, image $i$, ground-truth $gt$, two confidence thresholds $S_{high}$ and $S_{low}$, IoU threshold $\theta_{IoU}$
Ensure: loss $L_{frcnn}^{IncM}$
 1: Results of original model $boxes = NMS(OriM(i), \theta_{IoU})$
 2: Let pseudo ground-truth $boxes_p = \varnothing$
 3: for $box$ in $boxes$ do
 4:     if $IoU(box, gt) < \theta_{IoU}$ then
 5:         $boxes_p = boxes_p \cup box$
 6:     end if
 7: end for
 8: Compute $L_{RPN}^{IncM}$ with $S_{low}$ (Equation (9))
 9: Compute $L_{head}^{IncM}$ with $S_{high}$ (Equation (10))
10: Compute $L_{frcnn}^{IncM}$ (Equation (11))
11: return $L_{frcnn}^{IncM}$
```

both old and new classes, and (2) the detection head is trained to accurately discriminate between different classes. Therefore, we design a 2-threshold training strategy where a low threshold is used to select potential proposals for training the RPN and a high threshold is used to get the detection results with a high confidence for training a precise detection head.

Algorithm 1 describes the whole procedure of generating pseudo ground-truths and the 2-threshold training strategy for calculating the original Faster R-CNN losses. The losses of the original Faster R-CNN consist of the losses of the RPN and the detection head, which are calculated as follows:

$$L_{RPN}^{IncM} = loss(IncM.RPN((boxes_p > S_{low}) \cup gt)), \tag{9}$$

$$L_{head}^{IncM} = loss(IncM.head((boxes_p > S_{high}) \cup gt)), \tag{10}$$

$$L_{frcnn}^{IncM} = L_{RPN}^{IncM} + L_{head}^{IncM}, \tag{11}$$

where $boxes_p$ is the pseudo ground-truth. The losses of the RPN and the detection head include classification and regression losses. S_{low} and S_{high} are two confidence thresholds, which are used to select the pseudo ground-truth for training the RPN and the detection head, respectively.

The overall loss L_{all} applied to incrementally learn the new classes is presented as Equation (12), which is the sum of the standard Faster R-CNN losses and the proposed distillation losses. The Faster R-CNN losses of IncM are applied to all classes, where the pseudo ground-truth of the old classes and the ground-truth of the new classes are used for training. The Faster R-CNN losses of ResM are applied to new classes, where only the ground-truth of the new classes are used for training. The distillation losses are used to constrain the learning of IncM.

$$L_{all} = L_{frcnn}^{IncM} + L_{frcnn}^{ResM} + \lambda \times (D_{fea} + D_{res} + D_{cls}), \tag{12}$$

where λ is a trade-off between the original Faster R-CNN losses and the proposed distillation losses. We set $\lambda = 1$ in all experiments.

4 EXPERIMENTS AND EVALUATION

4.1 Experiment Setup

4.1.1 Datasets. The proposed method is evaluated on two object detection benchmarks PASCAL VOC 2007 and Microsoft COCO. VOC2007 has 20 object classes and consists of $5K$ images in its trainval subset and $5K$ images in its test subset. We use the test subset for evaluation. COCO has $80K$ images in the training subset and $40K$ images in the validation set for 80 object classes, and the minival (the first $5K$ images from the validation set) split is used for evaluation. The experiments are conducted on different numbers of classes. There are several settings for evaluating our method:

(1) Following the procedure described in the work of Shmelkov et al. [40], we take 19, 15, or 10 classes from VOC2007 sorted in alphabetical order as the old classes, and the remaining 1, 5, or 10 classes as the corresponding new classes. For COCO, we take the first 40 classes as the old classes and the remaining 40 classes as the new classes.
(2) We split the trainval set of VOC2007 and the training set of COCO into four groups: A, B, C, and D, and use the same setting as described in the work of Hao et al. [13]. For each group, images that only contain the objects of classes in this group are selected.

4.1.2 Evaluation Metrics. The metrics we use for the evaluation are **mean average precision (mAP)** at 0.5 IoU threshold for VOC2007, and mAP across different IoU thresholds from 0.5 to 0.95 for COCO. Our method is compared against object detector fine-tuning as well as other recent related works based on two-stage object detectors [6, 13, 23, 40]. We list the results of these methods reported in their original papers, which are evaluated under the same settings as our proposed method. The best results are in bold, and the second-best results are underlined.

4.1.3 Implementation Details. OriM 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. IncM is trained for 10 epochs with $lr = 0.0001$ and decays to 0.00001 after 5 epochs. The confidence and IoU thresholds for NMS are set to 0.5 and 0.3, respectively. For pseudo ground-truth filtering, θ_{IoU} is set to 0.3, and S_{low} and S_{high} are set to 0.1 and 0.9, respectively. For fair comparison, we use the same backbone for all compared methods as reported in their original papers, including ResNet-50 and ResNet-101 [14]. We conduct all experiments on a single NVIDIA GTX 2080Ti.

In our experiment descriptions, the following notations are used: $A()$ refers to the results of OriM, and RD-IOD refers to the results of our incremental learning method, which is trained on the base of $A()$. We use D_{fea} to denote the general feature distillation between OriM and IncM, and D_{res} to denote residual distillation on the feature space including D_{res}^{base} and D_{res}^{pool}. D_{cls} represents the joint distillation on the classification layers. The 2-th represents the proposed incremental learning with 2-threshold training strategy.

4.2 Addition of Multiple Classes at Once

In the first experiment, we evaluate the performance of our proposed method on VOC2007 when 1, 5, and 10 new classes are added at once each time. The per-category average precision on VOC2007 test subset results on these three settings are listed in Table 1. We also report the mAP of these methods trained on all classes as listed in the fourth subtable (up-bound models (1–20)).

For the first setting in this table, we show the test results on the performance of 19 old classes and 1 new class (tvmonitor) from VOC2007. We train OriM on the VOC2007 trainval subset with all data containing any of these 19 classes (A(1–19)). IncM is trained on the data of VOC2007 trainval subset containing “tvmonitor.” In our experiments, the first baseline method is fine-tuning, and we

Table 1. Per-Class Average Precision (%) on the VOC2007 Test Dataset

										1											
Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
A(1–19)	76.8	81.4	75.9	59.2	56.8	81.6	84.8	84.2	51.8	82.5	67.2	83.3	83.7	79.7	78.2	47.1	73.4	68.0	78.0	-	73.3
Fine-tuning	31.8	24.7	28.3	25.5	24.6	43.6	61.4	35.3	10.6	35.6	17.5	22.3	27.5	20.0	20.0	16.8	28.1	11.1	28.7	56.5	28.5
Shmelkov et al. [40]	69.4	79.3	69.5	57.4	45.4	78.4	79.1	80.5	45.7	76.3	64.8	77.2	80.8	77.5	70.1	42.3	67.5	64.4	76.7	62.7	68.3
Chen et al. [6]											68.3									<60.0	<68.3
Li et al. [23]	69.7	78.3	70.2	46.4	59.5	69.3	79.7	79.9	52.7	69.8	57.4	75.8	69.1	69.8	76.4	43.2	68.5	70.9	53.7	40.4	65.0
Hao et al. [12]	72.6	79.5	74.1	54.4	52.0	73.1	80.8	85.1	46.7	78.9	58.5	81.1	85.0	81.5	77.0	43.7	72.8	65.0	70.5	60.0	69.6
Hao et al. [13]	72.9	81.2	73.3	52.9	52.4	73.8	81.3	84.3	47.0	79.8	57.8	80.9	85.4	80.4	77.0	43.2	72.6	64.2	70.5	60.0	69.5
RD-IOD	73.7	80.3	73.7	61.6	56.6	80.6	85.8	85.0	52.2	80.6	63.5	82.0	83.9	80.5	77.9	47.0	73.5	66.5	77.1	54.5	**71.8**
										5											
Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
A(1–15)	77.5	79.0	74.4	60.4	58.1	76.0	84.9	84.8	51.2	76.0	65.7	83.2	84.1	79.1	78.2	-	-	-	-	-	74.2
Fine-tuning	54.1	50.3	47.8	32.7	21.1	51.6	71.1	64.6	19.2	48.0	47.6	52.8	61.2	46.1	42.5	37.2	55.6	57.0	63.0	63.3	49.3
Shmelkov et al. [40]	70.5	79.2	68.8	59.1	53.2	75.4	79.4	78.8	46.6	59.4	59.0	75.8	71.8	78.6	69.6	33.7	61.5	63.1	71.7	62.2	65.9
Hao et al. [12]	69.1	77.9	67.8	50.7	53.4	54.2	82.8	82.2	46.9	66.3	63.2	75.1	81.3	77.3	74.7	34.9	65.2	59.3	70.9	63.7	65.9
Hao et al. [13]	68.1	78.0	68.2	52.2	53.7	54.6	83.2	80.5	46.3	67.4	58.6	75.7	81.2	76.5	74.6	34.4	63.8	59.0	70.7	62.6	65.5
RD-IOD	74.4	79.1	74.6	57.2	58.1	74.2	83.6	85.3	53.5	74.7	65.2	81.1	83.5	80.8	76.7	32.4	66.9	62.7	73.0	59.8	**69.8**
										10											
Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
A(1–10)	90.4	90.8	90.6	90.6	86.7	87.4	90.4	89.2	87.6	77.8	-	-	-	-	-	-	-	-	-	-	88.1
Fine-tuning	52.7	27.1	41.9	30.1	15.4	40.8	46.9	60.4	13.0	40.5	57.6	70.9	78.8	70.4	75.8	38.7	65.0	63.4	70.0	64.0	51.2
Shmelkov et al. [40]	69.9	70.4	69.4	54.3	48.0	68.7	78.9	68.4	45.5	58.1	59.7	72.7	73.5	73.2	66.3	29.5	63.4	61.6	69.3	62.2	63.1
Li et al. [23]	71.7	81.7	66.9	49.6	58.0	65.9	84.7	76.8	50.1	69.4	67.0	72.8	77.3	73.8	74.9	39.9	68.5	61.5	75.5	72.4	**67.9**
Hao et al. [12]	69.2	54.6	68.2	56.1	52.8	70.4	81.3	70.6	37.8	63.1	57.3	75.2	80.5	72.7	74.9	37.9	68.2	63.4	69.9	63.4	64.4
Hao et al. [13]	70.6	51.5	68.9	54.3	52.4	69.0	80.5	70.0	34.1	62.5	59.6	74.5	78.7	72.1	75.1	37.0	66.2	62.6	72.4	63.5	63.8
RD-IOD	76.6	71.4	71.0	62.2	58.2	79.8	83.5	79.1	43.2	74.7	56.4	74.7	77.3	72.9	72.5	33.6	68.2	60.2	71.3	62.2	67.5
										Up-Bound Models (1–20)											
Shmelkov et al. [40]	70.2	77.9	70.4	54.1	47.4	78.9	78.6	79.8	50.8	75.9	65.6	78.0	80.5	79.1	76.3	47.7	69.3	65.6	76.8	73.9	69.8
Chen et al. [6]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	70.7
Li et al. [23]	77.8	85.0	82.9	62.1	64.4	74.7	86.9	87.0	56.0	76.5	71.2	79.2	79.1	76.2	83.8	53.9	73.2	67.4	77.7	78.7	74.7
RD-IOD	78.9	78.9	74.9	64.6	56.1	81.8	84.6	84.7	52.5	83.6	66.7	84.6	84.2	78.5	78.3	47.9	74.8	69.4	78.6	73.4	73.9

Comparisons are conducted under different settings when 1, 5, or 10 classes are added at once.

initialize IncM using the parameters of OriM. Unlike the original fine-tuning, which trains a new classification layer from scratch for a new task, we initialize the parameters in the classification layer of IncM with those of OriM obtained from the old classes, to preserve the knowledge learned. However, as can be seen from the first part in Table 1, fine-tuning achieves only 28.5% mAP on all classes when old classes are the majority, which should have caused catastrophic forgetting. As reported by Chen et al. in their original paper [6], "68.3" is the average precision of the old classes and "<60.0" means the precision of the new class ("tvmonitor") is lower than 60.0%. Therefore, the roughly estimated mAP on all classes is lower than 68.3%. We also reproduce the distillation methods used in the work of Hao et al. [12, 13]. They both use Kullback-Leibler divergence loss for the distillation on the output layers, and Hao et al. [13] also use mean squared error for feature distillation. It is important to note that our proposed method (RD-IOD) with various distillation losses and 2-threshold training strategy designed achieves the highest accuracy (71.8%) with a 3.5% increase compared to that of Shmelkov et al. [40] and Chen et al. [6] and a 2.2% increase compared to Hao et al. [12]. The effectiveness of our proposed method for mitigating catastrophic forgetting is therefore demonstrated.

For the second setting, we choose the first 15 classes as the old classes for training OriM (A(1–15)), and the remaining 5 classes are used for incremental learning. As shown in the second part in Table 1, although the performance of fine-tuning is improved with the increased number of new classes, the accuracy on old classes is still lower than our method by a large margin. The mAP of RD-IOD reaches 69.8%, and increases by about 3.9% compared to Shmelkov et al. [40]. Our method also outperforms the reproduced work of Hao et al. [12, 13] under the same setting as ours.

Our method is also evaluated on the case with adding more classes (e.g., 10 classes) as shown in the third part of Table 1. More precisely, OriM is trained on 10 classes first (A(1–10)), and IncM is trained on the remaining 10 new classes. The proposed method with all components (D_{fea} D_{res} D_{cls} 2-th) achieves 67.5% mAP with a 4.4% increase compared to Shmelkov et al. [40]. We also list

Table 2. Average Precision (%) on COCO Minival (First 5,000 Validation Images)

Method	mAP@0.5	mAP@[0.5, 0.95]
A(1–40)	54.4	32.5
Shmelkov et al. [40]	37.4	21.3
RD-IOD	**42.8**	**23.7**
A(1–80)	49.6	29.0

Comparisons are conducted when 40 classes are added at once.

the results of Li et al. [23], which are reported in their original paper under the same split setting on the VOC2007 dataset. Although the mAP of Li et al. [23] is slightly better than our method on the 10+10 setting, our method exceeds it by a large margin (6.8%) on the 19 + 1 setting, which demonstrates the effectiveness of our method at preserving the learned knowledge without using the old classes from the old dataset. Compared with other methods, the mAP (67.5%) of our method is closer to the mAP (73.9%) of our up-bound model trained on all classes assuming the old dataset is also available. The gap between Shmelkov et al. [40] and their up-bound model is 6.7%, and the gap between Li et al. [23] and their up-bound model is 6.8%, which are both worse than our method.

The results show that our method outperforms the reproduced distillation methods of Hao et al. [12, 13] under all settings, which demonstrates that our proposed two-level residual distillation is more effective than the originally used distillation for incremental object detection.

Some visualized detection results on VOC2007 are presented in Figure 7. The top two rows and bottom two rows show the results when 5 and 10 classes are added at once, respectively. The results show that our method achieves satisfactory detection performance on both old and new classes. The old classes (dog, chair, person) on the 15 + 5 setting and the old classes (airplane, bicycle, bottle, boat, bus, table) on the 10 + 10 setting can be detected precisely.

We also present the t-SNE visualization of the features of both old and new classes before the final output layers, as shown in Figure 8. Compared with the t-SNE visualization of the features generated from the fine-tuned model, the features of different object classes are still discriminative after the execution of our proposed incremental learning method. This further demonstrates that our method can mitigate catastrophic forgetting while maintaining the ability to learn new classes.

Finally, we also test the proposed method with all distillation losses on COCO, where 40 classes are old classes and the remaining 40 classes are new classes. The results are listed in Table 2. The performance of RD-IOD outperforms Shmelkov et al. [40] by a large margin with 5.4% improvement on mAP@0.5 and 2.4% on mAP@[0.5, 0.95], and the gap to the result obtained from training on the whole dataset (A(1–80)) is reduced. This demonstrates the effectiveness of the proposed method on a larger dataset with more classes.

4.3 Sequential Addition of Multiple Classes

In this experiment, we evaluate the performance of our method by adding classes sequentially for incremental learning. IncM is updated on the basis of the latest trained network with a set of new classes, and the process is repeated with another set of new classes. We split the trainval set of VOC2007 and the training set of COCO into four groups, A, B, C, and D as described in the work of Hao et al. [13], and each group contains 5 (VOC2007) or 20 (COCO) classes. In this scenario, the images in each group only contain the objects of classes in this group, which ensures that only

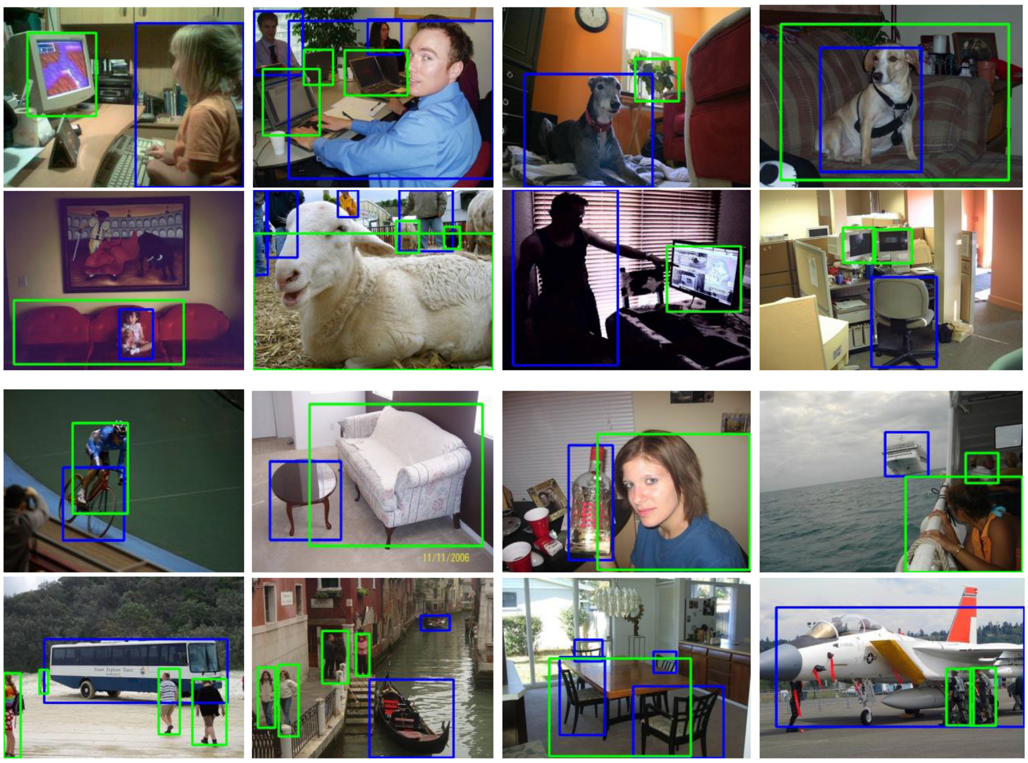


Fig. 7. Some visualized results on VOC2007 when 5 and 10 classes are added at once (blue: old classes, green: new classes). The top two rows and bottom two rows represent the detection results when 5 and 10 classes are added at once, respectively.

Table 3. Average Precision (%) on the VOC2007 Test Dataset and the COCO Validation Dataset, When Four Groups Are Added Sequentially

Method	VOC2007					COCO				
	A	B	C	D	mAP	A	B	C	D	mAP
Shmelkov et al. [40]	68.8	-	-	-	68.8	65.0	-	-	-	65.0
	46.5	70.4	-	-	58.5	32.6	20.4	-	-	26.5
	14.5	9.9	51.6	-	25.3	12.5	5.6	24.1	-	14.1
	4.6 (−64.3)	1.3 (−69.1)	22.4 (−29.2)	38.8	16.8 (−52.1)	6.5 (−58.5)	1.7 (−18.7)	3.1 (−21)	23.3	8.6 (−56.4)
Hao et al. [13]	68.8	-	-	-	68.8	65.0	-	-	-	65.0
	60.5	69.5	-	-	65.0	41.7	21.6	-	-	31.7
	25.2	28.1	54.3	-	35.9	29.1	9.5	24.2	-	21.0
	10.2 (−58.7)	4.9 (−64.6)	31.9 (−22.4)	39.6	21.6 (−47.3)	11.4 (−53.6)	4.6 (−17)	2.8 (−21.4)	28.5	11.8 (−53.2)
RD-IOD	68.8	-	-	-	68.8	65.0	-	-	-	65.0
	60.7	68.3	-	-	64.5	56.5	21.8	-	-	39.1
	49.4	47.5	57.8	-	51.6	52.0	15.3	18.9	-	28.7
	40.1 (−28.7)	36.1 (−32.2)	44.5 (−13.3)	37.9	**39.6 (−29.2)**	43.2 (−21.8)	12.5 (−9.3)	12.3 (−6.6)	25.7	**23.4 (−41.6)**

images of the new classes are utilized to perform class-incremental learning. ResNet101 is used in this experiment for fair comparison.

For a fair comparison, we reproduce the distillation methods [13, 40] as shown in Table 3. All experiments are done under exactly the same setting as our proposed method. The sequential additions are conducted on the same baseline with our method. As can be seen, our method also achieves promising results compared with other methods. The mAP of RD-IOD exceeds other methods in almost all incremental learning steps. Due to the pesudo ground-truth mechanism,

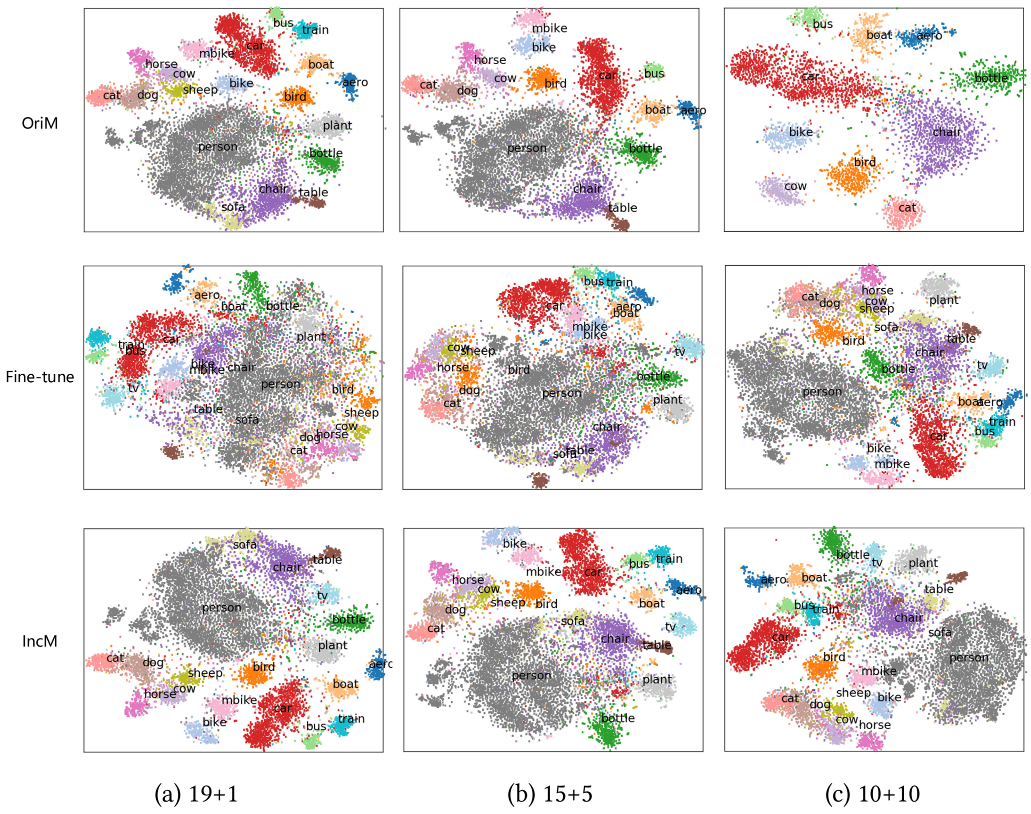


Fig. 8. t-SNE visualization of the features of both old and new classes before the final output layers under three incremental settings on the VOC2007 test dataset. The three rows show the features from (1) the original model, (2) the fine-tuned model, and (3) the incremental model, respectively.

we conclude that our method is more appropriate for the scenarios, in which some objects of the old classes exist in the new data without annotations. In these experiments, it can be seen that our method also performs well on overcoming catastrophic forgetting. We list the decreased mAP ("–") of each group in the last learning stage compared with the mAP in the first learning stage. After the fourth learning step, our method outperforms other methods, which demonstrates the effectiveness of preserving the learned knowledge and incrementally learning new object classes.

4.4 Ablation Study

To demonstrate the effectiveness of the key components, we conduct experiments to evaluate them separately when 1, 5, and 10 classes are added on VOC2007 at once each time. Results are shown in Table 4 where the first row represents the model trained on the new classes with the original losses of Faster R-CNN and the pseudo ground-truth generated from OriM using a confidence threshold of 0.5. The following four rows show the results of adding the designed losses and 2-threshold training strategy separately. The "+" sign indicates an increased mAP when compared with the first row. As listed in the table, the base feature distillation D_{fea} improves by 0.67% when only one class is added. This verifies the effectiveness in preserving the performance on old classes. The performance of D_{fea}, when used alone, is slightly decreased with the increasing number of

Table 4. Ablation Study

Components				mAP(%)		
D_{fea}	D_{res}	D_{cls}	$2\text{-}th$	1	5	10
				69.10	66.05	64.08
√				69.77 (+0.67)	66.01 (−0.03)	63.94 (−0.14)
	√			71.42 (+2.32)	68.99 (+2.94)	65.03 (+0.95)
		√		69.34 (+0.24)	66.08 (+0.03)	64.55 (+0.47)
			√	69.63 (+0.52)	66.40 (+0.35)	66.05 (+1.97)
√	√	√	√	**71.82** (+2.72)	**69.84** (+3.79)	**67.45** (+3.37)
	√	√	√	71.48 (−0.34)	69.48 (−0.36)	67.17 (−0.28)
√		√	√	69.79 (−2.03)	67.92 (−1.92)	66.58 (−0.87)
√	√		√	71.42 (−0.40)	69.46 (−0.38)	67.07 (−0.38)
√	√	√		71.43 (−0.39)	68.97 (−0.87)	65.07 (−2.38)

Table 5. Two-Level Residual Distillation Losses

Method		1	5	10
RD-IOD		71.82	69.84	67.45
D_{res}	w/o D_{res}^{base}	71.72 (−0.10)	69.71 (−0.13)	67.26 (−0.19)
	w/o D_{res}^{pool}	70.43 (−1.39)	67.36 (−2.48)	66.79 (−0.66)

Table 6. Comparisons on Alternative Distillation Losses

Method		1	5	10
D_{fea}	L1 Loss	71.14	68.17	62.88
	NDL	**71.43** (+0.29)	**68.97** (+0.80)	**65.07** (+2.19)
D_{res}^{pool}	NDL	68.65	65.94	63.85
	L1 Loss	**71.82** (+3.17)	**69.84** (+3.90)	**67.45** (+3.60)
D_{cls}	Old	69.31	65.75	64.08
	Old&new	**69.34** (+0.03)	**66.08** (+0.33)	**64.55** (+0.47)

new classes D_{fea}, because it is designed for preserving the performance on the old classes, and it requires other components for improving the mAP of all classes. D_{res} increases by about 2.07% on average, and the joint distillation of the final classification layer D_{cls} increases by about 0.25% on average when used alone. The 2-threshold training strategy ($2\text{-}th$) is also effective for boosting the performance, which increases by about 0.95% on average.

We also evaluate these components by removing them individually as shown in the last four rows of Table 4, where "–" represents the decreased mAP compared with the combination of all components. As can be seen, all components play important roles in our method. The mAP decreases by about 0.33% on average without D_{fea}, and the mAP decreases by about 1.61% on average without D_{res}, which demonstrates the importance of the residual distillation. In addition, the combinations without D_{cls} or $2\text{-}th$ also lead to performance degradation. Table 5 verifies the importance of residual distillation at the image level and the instance level, respectively. As demonstrated in the results, both of the two levels of residual distillation have an impact on the accuracy, especially D_{res}^{pool}.

The comparison of alternative distillation losses is shown in Table 6. The experiments are conducted on three settings: adding 1, 5, and 10 new classes, respectively. To evaluate the designed NDL feature distillation D_{fea}, we replace the loss function between two 2D feature maps of

Table 7. The mAP with Different Choices of Confidence Thresholds for Training RD-IOD

Confidence Threshold		1		5		10	
		Old	All	Old	All	Old	All
1-threshold	0.1	70.13	69.43	69.49	66.64	59.94	61.85
	0.5	72.22	71.43	72.31	68.97	64.78	65.07
	0.9	72.43	71.64	73.14	69.51	68.78	67.06
2-threshold	0.1&0.9	**72.73**	**71.82**	**73.47**	**69.84**	**69.96**	**67.45**
	0.5&0.9	72.22	71.46	72.96	69.62	69.03	66.88

Table 8. Comparisons Between the Jointly Trained and Pre-Trained Residual Models Applied to New Classes

Method	1	5	10
Pre-trained ResM	66.94	68.87	67.58
RD-IOD	71.82 (+4.88)	69.84 (+0.97)	67.45 (−0.13)

backbones in D_{fea} and D_{res} with L1-loss, which is directly applied onto the original 2D feature maps. The mAP of the designed NDL feature distillation exceeds L1-loss by around 1.09% on average in all three settings. To be consistent with D_{res}^{base}, we also replace L1-loss with NDL in D_{res}^{pool}. The results show that when L1-loss is directly applied onto the instance-level feature map, it yields a more effective result. For D_{cls}, the classification distillation from both OriM and ResM is compared with the distillation from only OriM. The joint classification distillation from both OriM and ResM outperforms the distillation from only OriM with the mAP increasing by about 0.28%. The results verify that our designs are more appropriate in this scenario.

To verify the effectiveness of the 2-threshold training strategy for maintaining the performance on old classes, we conduct comparisons between the 1-threshold strategy and 2-threshold strategy for training RD-IOD on three settings that have more old classes than the new ones. The results are illustrated in Table 7, where we present the mAP on both all classes and old classes. As can be seen, the 2-threshold choice with 0.1 and 0.9 can maintain the performance of old classes to a large extent and the mAP of all classes is also the highest compared with other choices for the confidence threshold.

In our residual distillation mechanism, ResM is jointly trained to detect new classes in the incremental learning stage. We compare its effectiveness with a strategy that uses a pre-trained ResM on new classes to assist the learning of IncM. We evaluate these ideas on $19 + 1$, $15 + 5$, and $10 + 10$ class settings as shown in Table 8. Because in the pre-trained ResM, the parameters are frozen in the incremental learning stage, the capability of ResM to fit the difference between OriM and IncM is suppressed. We believe that ResM should learn to guide the residual features between OriM and ResM toward the features of the new classes and strike a balance between detecting the new classes and fitting the residual features. Our intuition is confirmed by the experiment results: in the $19 + 1$ setting, the joint training strategy outperforms the pre-trained ResM in terms of mAP by 4.88%. Even though the performance of the pre-trained ResM is slightly improved when the number of the new classes is increased, as shown in the 10 + 10 setting, the training time is also increased because of the extra learning step added to pre-training ResM. Therefore, it is evident that joint training of ResM is a better choice for the task at hand.

We also list the training time under various situations and the GPU memory usage of our proposed method and training from scratch as shown in Table 9. This table includes the training

Table 9. Training Time (in Seconds) and GPU Memory (in Megabytes)

Method	1	5	10	GPU Memory
RD-IOD	3,048.35	12,586.75	35,911.68	6,498
From scratch		36,006.98		5,090

Table 10. RD-IOD on RetinaNet, Cascade R-CNN, and SSD

Method		aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
RetinaNet	A(1–15)	72.4	81.5	77.6	57.2	58.1	73.1	83.8	83.9	54.2	65.3	61.0	81.4	80.1	71.6	80.4	-	-	-	-	-	72.1
	Fine-tune	27.1	10.9	39.1	11.5	16.5	55.8	64.8	35.0	11.3	24.8	7.5	43.0	40.1	51.1	23.0	33.7	40.8	68.1	40.9	67.6	35.6
	RD-IOD w/o D_{res}	68.7	75.0	70.0	48.6	50.6	67.8	81.7	78.9	44.0	54.8	49.1	72.4	74.8	68.1	73.6	14.8	21.0	41.1	16.4	32.5	55.2
	RD-IOD	67.4	77.6	74.6	47.1	50.9	67.4	80.7	76.2	44.5	57.7	48.9	75.9	78.1	70.5	77.5	16.9	22.7	46.6	18.7	39.5	**57.0**
Cascade R-CNN	A(1–15)	79.1	80.1	76.1	60.7	59.0	76.8	80.7	78.6	51.9	70.3	62.8	86.5	80.4	78.3	78.8	-	-	-	-	-	73.3
	Fine-tune	17.6	0.0	9.1	9.1	0.0	4.5	9.1	0.0	0.0	0.0	0.0	0.0	17.9	9.1	0.0	39.3	46.4	64.6	49.9	69.3	17.3
	RD-IOD w/o D_{res}	54.2	79.7	69.3	46.8	58.0	61.7	80.3	77.6	49.0	72.1	58.6	78.5	79.5	70.5	77.8	20.7	44.9	59.4	50.2	47.4	61.8
	RD-IOD	62.7	71.6	69.2	48.0	51.0	74.7	80.2	77.0	47.4	74.9	60.7	78.8	79.4	70.5	77.9	19.1	49.4	57.4	58.0	45.3	**62.7**
SSD	A(1–15)	74.9	79.0	65.8	56.9	41.7	74.1	83.8	78.8	51.4	60.6	70.7	72.6	83.2	78.7	74.3	-	-	-	-	-	69.8
	Fine-tune	9.1	3.6	9.1	9.1	9.1	4.5	9.1	9.1	4.5	9.1	0.0	9.1	0.0	9.1	9.1	28.3	29.4	55.9	39.0	57.3	15.7
	RD-IOD w/o D_{res}	59.6	77.0	57.2	51.3	28.7	64.7	78.9	74.8	49.9	49.7	64.4	71.1	78.9	70.6	67.4	23.1	33.2	58.2	57.9	57.8	58.7
	RD-IOD	66.3	75.8	56.5	52.4	33.8	63.3	78.7	75.7	53.6	48.7	65.1	69.8	78.6	69.0	71.0	22.4	36.2	62.3	62.1	62.1	**60.2**

Per-class average precision (%) on the VOC2007 test dataset.

time of (1) executing our proposed method when using only new data, and (2) training the model from scratch using both old and new data. The training time of our method with only new classes is much less than training from scratch when the number of new classes is much less than old classes, such as 19 + 1 and 15 + 5 settings. The inference time remains similar because the model structure in the inference procedure is the same as the original Faster R-CNN except in the final output layers. OriM and ResM are just used to assist the learning of IncM. We only use one NVIDIA 2080Ti for all experiments. The GPU memory usage of RD-IOD (6,498M) is acceptable compared to training from scratch (5,090M).

4.5 Generalization

The proposed RD-IOD can be considered as a general distillation method to preserve the learned knowledge in the original model and incrementally learn to detect new classes simultaneously for incremental object detection. This suggested method applied to the feature space can be easily adapted to other object detectors. To verify this generalization, we applied the proposed method to RetinaNet [25], Cascade R-CNN [4], and SSD [28], the representative one-stage and two-stage object detection frameworks. As shown in Table 10, our proposed method outperforms the original RetinaNet and SSD on fine-tuning by a large margin. This demonstrates that our method can effectively mitigate catastrophic forgetting on one-stage object detection as well as on two-stage object detection. Compared to without the residual distillation, the three performance indicators of RD-IOD with residual distillation is consistently improved, which demonstrates the generalization of RD-IOD and its effectiveness for better incrementally learning new classes.

5 CONCLUSION

In this article, we propose a triple-network-based incremental object detector with a novel residual distillation scheme for learning new classes without using original training data. A frozen copy of the original model trained on data of old classes is used to generate pseudo ground-truth with a 2-threshold strategy and to provide knowledge corresponding to old classes for training the incremental model. A residual model is jointly trained on new classes to preserve the discrimination between old and new classes by learning the residual features obtained from the incremental model and the original model. A two-level residual distillation loss is designed in both image level and instance level, and a joint classification distillation is designed for the output layers. Experimental results on VOC2007 and COCO demonstrate the effectiveness of the proposed method on incrementally learning to detect objects of new classes without forgetting the original learned knowledge. In future work, to better maintain feature discrimination in the incremental learning procedure, we plan to combine self-supervised/contrastive learning [19, 47] with incremental object detection.

REFERENCES

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

[2] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. 2017. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3366–3375.

[3] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T. Barron, Ferran Marques, and Jitendra Malik. 2014. Multiscale combinatorial grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 328–335.

[4] Zhaowei Cai and Nuno Vasconcelos. 2018. Cascade R-CNN: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6154–6162.

[5] Gert Cauwenberghs and Tomaso Poggio. 2001. Incremental and decremental support vector machine learning. In *Advances in Neural Information Processing Systems*. 409–415.

[6] Li Chen, Chunyan Yu, and Lvcai Chen. 2019. A new knowledge distillation for incremental object detection. In *Proceedings of the 2019 International Joint Conference on Neural Networks*. IEEE, Los Alamitos, CA, 1–7.

[7] Yudi Chen, Wei Wang, Yu Zhou, Fei Yang, Dongbao Yang, and Weiping Wang. 2021. Self-training for domain adaptive scene text detection. In *Proceedings of the 2020 25th International Conference on Pattern Recognition*. IEEE, Los Alamitos, CA, 850–857.

[8] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. 2010. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision* 88, 2 (2010), 303–338.

[9] Robert M. French. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences* 3, 4 (1999), 128–135.

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

[11] Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211* (2013).

[12] Yu Hao, Yanwei Fu, and Yu-Gang Jiang. 2019. Take goods from shelves: A dataset for class-incremental object detection. In *Proceedings of the 2019 International Conference on Multimedia Retrieval*. 271–278.

[13] Yu Hao, Yanwei Fu, Yu-Gang Jiang, and Qi Tian. 2019. An end-to-end architecture for class-incremental object detection with knowledge distillation. In *Proceedings of the 2019 IEEE International Conference on Multimedia and Expo*. IEEE, Los Alamitos, CA, 1–6.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

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

[16] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. 2019. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 831–839.

[17] Heechul Jung, Jeongwoo Ju, Minju Jung, and Junmo Kim. 2016. Less-forgetting learning in deep neural networks. *arXiv preprint arXiv:1607.00122* (2016).

[18] Heechul Jung, Jeongwoo Ju, Minju Jung, and Junmo Kim. 2018. Less-forgetful learning for domain expansion in deep neural networks. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. 3358–3365.

[19] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems 33*.

[20] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences* 114, 13 (2017), 3521–3526.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. 1097–1105.

[22] Ilja Kuzborskij, Francesco Orabona, and Barbara Caputo. 2013. From n to n+1: Multiclass transfer incremental learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3358–3365.

[23] Dawei Li, Serafettin Tasci, Shalini Ghosh, Jingwen Zhu, Junting Zhang, and Larry Heck. 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.

[24] Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 12 (2017), 2935–2947.

[25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 2980–2988.

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*. 740–755.

[27] Chuanbin Liu, Hongtao Xie, Zhengjun Zha, Lingyun Yu, Zhineng Chen, and Yongdong Zhang. 2019. Bidirectional attention-recognition model for fine-grained object classification. *IEEE Transactions on Multimedia* 22, 7 (2019), 1785–1795.

[28] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*. 21–37.

[29] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. 2020. Video cloze procedure for self-supervised spatio-temporal learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11701–11708.

[30] Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*. Vol. 24. Elsevier, 109–165.

[31] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. 2013. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 11 (2013), 2624–2637.

[32] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy Hospedales, and Tao Xiang. 2020. Incremental few-shot object detection. *arXiv preprint arXiv:2003.04668* (2020).

[33] Robi Polikar, Lalita Upda, Satish S. Upda, and Vasant Honavar. 2001. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systems, Man, and Cybernetics: Part C (Applications and Reviews)* 31, 4 (2001), 497–508.

[34] Zhi Qiao, Yu Zhou, Dongbao Yang, Yucan Zhou, and Weiping Wang. 2020. Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 13528–13537.

[35] Xugong Qin, Yu Zhou, Youhui Guo, Dayan Wu, and Weiping Wang. 2021. FC 2 RN: A fully convolutional corner refinement network for accurate multi-oriented scene text detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, Los Alamitos, CA, 4350–4354.

[36] Xugong Qin, Yu Zhou, Dongbao Yang, and Weiping Wang. 2019. Curved text detection in natural scene images with semi-and weakly-supervised learning. In *Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR'19)*. IEEE, Los Alamitos, CA, 559–564.

[37] Amal Rannen, Rahaf Aljundi, Matthew B. Blaschko, and Tinne Tuytelaars. 2017. Encoder based lifelong learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 1320–1328.

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

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*. 91–99.

[40] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. 2017. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE International Conference on Computer Vision*. 3400–3409.

[41] Gan Sun, Yang Cong, and Xiaowei Xu. 2018. Active lifelong learning with "Watchdog." In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. 4107–4114.

[42] Gan Sun, Cong Yang, Ji Liu, Lianqing Liu, Xiaowei Xu, and Haibin Yu. 2018. Lifelong metric learning. *IEEE Transactions on Cybernetics* 49, 8 (2018), 3168–3179.

[43] Yuxin Wang, Hongtao Xie, Zheng-Jun Zha, Youliang Tian, Zilong Fu, and Yongdong Zhang. 2020. R-Net: A relationship network for efficient and accurate scene text detection. *IEEE Transactions on Multimedia* 23 (2020), 1316–1329.

[44] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. 2020. Video playback rate perception for self-supervised spatio-temporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6548–6557.

[45] Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning—Volume 70*. 3987–3995.

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

[47] Yifei Zhang, Chang Liu, Yu Zhou, Wei Wang, Weiping Wang, and Qixiang Ye. 2021. Progressive cluster purification for unsupervised feature learning. In *Proceedings of the 2020 25th International Conference on Pattern Recognition.* 8476–8483.

[48] Bineng Zhong, Bing Bai, Jun Li, Yulun Zhang, and Yun Fu. 2018. Hierarchical tracking by reinforcement learning-based searching and coarse-to-fine verifying. *IEEE Transactions on Image Processing* 28, 5 (2018), 2331–2341.

[49] Qinqin Zhou, Bineng Zhong, Xiangyuan Lan, Gan Sun, Yulun Zhang, Baochang Zhang, and Rongrong Ji. 2020. Fine-grained spatial alignment model for person re-identification with focal triplet loss. *IEEE Transactions on Image Processing* 29 (2020), 7578–7589.

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

[51] C. Lawrence Zitnick and Piotr Dollár. 2014. Edge boxes: Locating object proposals from edges. In *Proceedings of the European Conference on Computer Vision.* 391–405.

[52] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. 2019. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE International Conference on Computer Vision.* 8420–8429.

[53] Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E. Gonzalez, and Fisher Yu. 2020. Frustratingly simple few-shot object detection. *arXiv preprint arXiv:2003.06957.*

[54] Dingwen Zhang, Haibin Tian, and Jungong Han. 2021. Few-cost salient object detection with adversarial-paced learning. *arXiv preprint arXiv:2104.01928.*

Received September 2020; revised May 2021; accepted June 2021