Class-Incremental Instance Segmentation via Multi-Teacher Networks

Yanan Gu, Cheng Deng*, and Kun Wei
School of Electronic Engineering, Xidian University, Xi'an 710071, China
{yanangu.xd, chdeng.xd,weikunsk}@gmail.com

Abstract

Although deep neural networks have achieved amazing results on instance segmentation, they are still ill-equipped when they are required to learn new tasks incrementally. Concretely, they suffer from "catastrophic forgetting", an abrupt degradation of performance on old classes with the initial training data missing. Moreover, they are subjected to a negative transfer problem on new classes, which renders the model unable to update its knowledge while preserving the previous knowledge. To address these problems, we propose an incremental instance segmentation method that consists of three networks: Former Teacher Network (FTN), Current Student Network (CSN) and Current Teacher Network (CTN). Specifically, FTN supervises CSN to preserve the previous knowledge, and CTN supervises CSN to adapt to new classes. The supervision of two teacher networks is achieved by a distillation loss function for instances, bounding boxes, and classes. In addition, we adjust the supervision weights of different teacher networks to balance between the knowledge preservation for former classes and the adaption to new classes. Extensive experimental results on PASCAL 2012 SBD and COCO datasets show the effectiveness of the proposed method.

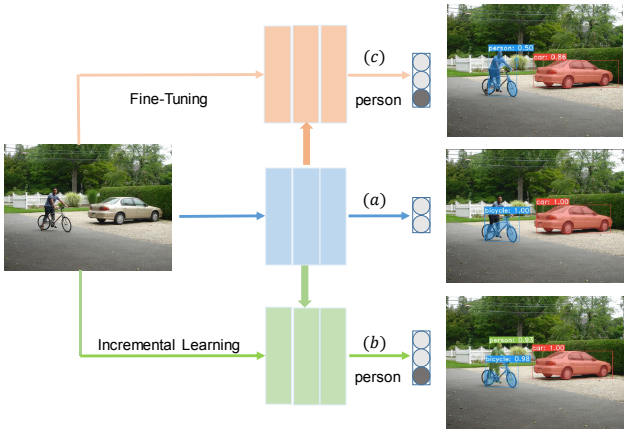


Figure 1: Comparison of incremental learning and fine-tuning. An instance segmentation model is firstly trained on two classes, including car and bicycle. Then it is retrained with images of the new class person.

Introduction

Instance segmentation is one of the fundamental tasks in computer vision, which is challenging because it not only needs to detect all objects in an image correctly but also needs to segment each instance precisely. Thanks to the development of deep learning, like other visual tasks (Wei et al. 2019; Yang et al. 2019; Han et al. 2020; Feng et al. 2020), current instance segmentation methods (He et al. 2017; Bolya et al. 2019; Chen et al. 2020) based on convolutional neural networks (CNNs) have achieved remarkable results. However, a fatal limitation of these methods lies in assuming the training data for all categories are always available, making them unsuitable in some real-world situations.

In some real-world situations, training data are received sequentially. The training data for the new task are available, but the data for the previous task are often not accessible due to various problems such as storage budget and privacy. A well-qualified incremental system needs to be upgraded continuously by absorbing knowledge from new tasks while preserving initially learned capabilities. Therefore, there are two main aspects to judge whether an incremental model is good or not: the ability to maintain knowledge of the previous tasks and capture new knowledge from the current task.

For the first aspect, the critical challenge is catastrophic forgetting, an abrupt degradation of performance on the original set of classes, when the training objective is adapted to the new classes without access to previous training data. For the second aspect, the critical challenge is the inability of a model to update its knowledge while preserving the previous knowledge, which is a special negative transfer problem. Recently, many studies on incremental schemes (Li and Hoiem 2017; Chaudhry et al. 2018; Shmelkov, Schmid, and Alahari 2017; Yu et al. 2020) for image classification or object detection have been proposed. However, catastrophic forgetting and negative transfer problem in incremental instance segmentation algorithms are still ignored and unsolved.

Figure 1 shows an incremental situation. We first train an instance segmentation model on two classes, including car and bicycle. As shown in Figure 1(a), the model segments

*Corresponding Author

car and bicycle clearly. Then the new class person is added to the training. Figure 1(c) shows a fine-tuning (Girshick et al. 2014) result on new class data. We can see that the bicycle is lost in the segmentation result, and the confidence score of the car becomes lower than before. Besides, the segmentation of the new class person is not only low in confidence score but also inaccurate. Unlike fine-tuning, we want all three classes to be segmented accurately in the final result. As shown in Figure 1(b), the old classes are preserved, and the new class is segmented precisely.

Using only the training data for the new classes, we propose an incremental instance segmentation method, which utilizes multi-teacher networks to supervise the model to preserve old knowledge and adapt to the new classes. The concept of multi-teacher networks is inspired by the human learning process (Hou et al. 2018). If a person wants to have good academic performance, he needs the guidance of multiple excellent teachers. Under the guidance of multiple teachers with different labor divisions, he can learn new knowledge better without forgetting the knowledge learned before. In our method, we introduce a Former Teacher Network (FTN) to help the Current Student Network (CSN) to preserve the previous knowledge and a Current Teacher Network (CTN) to help CSN adapt to new classes. Specifically, we utilize knowledge distillation to simulate the guidance process, which is proposed in (Hinton, Vinyals, and Dean 2015) and widely used in incremental tasks (Li and Hoiem 2017; Hou et al. 2018; Wei, Deng, and Yang 2020) and other tasks (Yang et al. 2020c). The core of the knowledge distillation in our method is to minimize the discrepancy between teacher and student networks responses. In instance segmentation task, the responses can be divided into three aspects: instance, bounding box, and classification. Therefore, we propose a distillation loss function that includes three corresponding distillation losses. Besides, we also seek a balance between those distillation losses to achieve better segmentation precision. What's more, the supervision is performed on the learning of both old and new classes, the balance of supervision between different teachers is also considered. Experiment results on PASCAL 2012 SBD and COCO dataset demonstrate our method can segment the new classes well while preserving the ability to segment old classes.

In summary, the major contributions of this paper are three folds:

• We propose an algorithm to achieve instance segmentation in incremental scenarios. This is the first attempt that applies incremental setting in instance segmentation task to the best of our knowledge.

• We formulate our proposed incremental instance segmentation method with multi-teacher networks to balance between the knowledge preservation for former classes and the adaption to new classes.

• Extensive experimental results on PASCAL 2012 SBD and COCO datasets prove the effectiveness of the proposed method. And we evaluate variants of our method with ablation studies to verify the effectiveness of each component in our model.

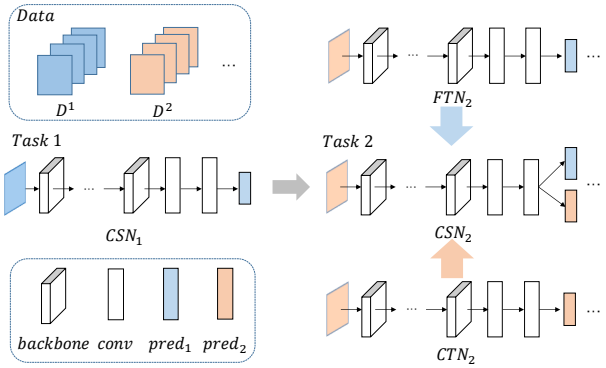


Figure 2: The incremental learning process of our method.

Related Work

Incremental Learning

A variety of incremental learning strategies (Li and Hoiem 2017; Dhar et al. 2019; Zhao et al. 2021) have been explored to prevent models from forgetting previously learned tasks. Li et al. proposed Learning with Forgetting (LwF) (Li and Hoiem 2017), which uses the new data to supervise the learning of the new tasks and to provide unsupervised output guidance on the previous tasks to prevent catastrophic forgetting. Further, Dhar et al. introduced Learning without Memorizing (Dhar et al. 2019), which extends LwF by adding a distillation term based on attention maps. Inspired by bayesian learning, Elastic Weight Consolidation (EWC) (Kirkpatrick et al. 2016) remembers previous tasks by selectively slowing down the learning on the important weights for those tasks. Specifically, it estimates the fisher matrix to weight a regularization term to constrain important parameters to stay close to their old values. Serra et al. proposed Hard Attention to the Task (HAT) (Serra et al. 2018) to learn hard attention masks concurrently to every task. The attention vectors of previous models are used to define a mask and constrain the updates of the weights of current models.

Recently, the more challenging class-incremental setting has attracted more attention. The difficulty of this setting is that the network has no access to the class-ID at the inference phase, which makes this setting more challenging. Our method uses this challenging setting, and the incremental learning process is performed on the same dataset.

In addition, different from the image classification task, it is widespread for the new classes and the old classes to appear in the same picture simultaneously in instance segmentation, which may confuse the network. For example, the new class that appeared in the previous task is regarded as the background class and is regarded as its genuine class in the current task, which increases the difficulty of incremental learning in instance segmentation.

Instance Segmentation

Instance segmentation is an essential task in computer vision, which enables numerous downstream vision applications. Due to the development of deep learning, like other

visual tasks (Yang et al. 2018, 2020a; Dang et al. 2020; Yang et al. 2020b; Deng et al. 2019), instance segmentation methods have achieved amazing results. Specifically, the current instance segmentation methods are mainly divided into two categories: two-stage methods and one-stage methods.

Two-stage methods formulate this task as the paradigm of "Detect then Segment" (Li et al. 2017; He et al. 2017; Liu et al. 2018; Chen et al. 2019). These methods first detect the objects and then predict the foreground masks on each bounding box. Mask R-CNN (He et al. 2017), built upon Faster R-CNN (Ren et al. 2015), extends the original detector by adding a branch for segmenting the instances within the detected bounding boxes. In addition, ROI-Pooling (Girshick 2015) is replaced by ROI-Align, which contributes to the improvement of accuracy. Based on Mask R-CNN, PANet (Liu et al. 2018) introduces bottom-up path augmentation, adaptive feature pooling, and fully-connected fusion to enhance the feature representation to improve the accuracy of instance segmentation. Further, Mask Scoring R-CNN adds a mask-IoU branch to re-score the confidence of the mask from the classification score. In summary, two-stage methods first detect bounding boxes and then segment in each bounding box. They can achieve the most advanced performance but are generally slower.

One-stage methods (Dai et al. 2016; Liu et al. 2018; Bolya et al. 2019; Chen et al. 2018) are conceptually faster than the two-stage methods. InstanceFCN (Dai et al. 2016) first produces some instance-sensitive score maps, then it uses an assembling module to generate object instances in a sliding window. Using keypoint detection to predict eight extreme points of one instance and generate an octagon mask, ExtremeNet (Zhou, Zhuo, and Krahenbuhl 2019) can achieve relatively reasonable object mask prediction. Instead of using position-controlled tiles, YOLACT (Bolya et al. 2019) first generates a set of prototype masks and learns a set of linear combination coefficients for each instance alongside the box predictions. It linearly combines the prototypes using the corresponding predicted coefficients and then crops with a predicted bounding box to generate the final mask. Since this process does not depend on repooling, YOLACT can produce high-quality masks and show temporal stability.

The above methods assume that the training data for all categories are always available, which is unsuitable in incremental situations. Different from those methods, we propose an incremental instance segmentation method that performs well in incremental situations. Specifically, considering the computation of category-specific proposals (Shmelkov, Schmid, and Alahari 2017) in two-stage methods like Mask R-CNN (He et al. 2017), we choose the one-stage method YOLACT (Bolya et al. 2019) as our incremental base instance segmentation model, because the prior boxes in YOLACT are agnostic to object categories.

Transfer Learning

Our work also involves transfer learning methods. Transfer learning focuses on storing knowledge gained while solving one task and applying it to a different but related task. Fine-tuning is a special case of transfer learning, which is popular in computer vision. The network trained on ImageNet for image classification (Deng et al. 2009) is often used as the basic model to train other tasks like semantic segmentation and object detection (Zitnick et al. 2014; Oquab et al. 2014).

However, there are some issues in Transfer Learning, such as "Negative Transfer". This problem is that the transferred knowledge will damage the performance of the target domain. This problem also exists in our setting.

Unlike the traditional negative transfer problem, which is often caused by the large difference between the target domain and the source domain, there are two main reasons for the negative transfer of our method. Firstly, since the current task model needs to calculate the distillation loss with the model of the previous task to preserve the old knowledge, the optimization space of parameters in the current model becomes smaller, limiting the learning ability of the current model on new classes. Secondly, to better retain the old knowledge, the learning rate in the training process of the new class is reduced, which also limits the learning of new classes. To solve this problem, we introduce a current teacher network, which only considers the learning of the new classes. A good current teacher network can give the current task a good initialization and soft supervision, which leads to a more gentle learning curve for the learning of the new classes.

Proposed Method

Problem Definition

We consider a class-incremental learning scenario. The model learns a series of tasks, each task contains several new classes. Given a dataset $D = \{(x, y)|x \in \mathcal{X}, y \in \mathcal{Y}\}$, where $\mathcal{X}$ represents images and $\mathcal{Y}$ represents labels. The number of total classes is $\mathcal{C}$. Given T tasks, we split $\mathcal{C}$ into T subsets $\mathcal{C}^1, \mathcal{C}^2, \ldots, \mathcal{C}^T$. We define task t as introducing new classes $\mathcal{C}^t$ using dataset $D^t = \{(x, y)|y \in \mathcal{C}^t\}$. The training images and labels are defined as $\mathcal{X}^t = \{x|(x, y) \in \mathcal{D}^t\}$ and $\mathcal{Y}^t = \{y|(x, y) \in \mathcal{D}^t\}$, respectively. Especially, $\mathcal{Y}^i \bigcap \mathcal{Y}^j = \emptyset$ for $i \neq j$. The goal of task T is to train a model that can segment the objects belonging to class $\mathcal{C}^t$, while still preserving the ability to segment the objects belonging to class $\mathcal{C}^i, i < t$.

Instance Segmentation Network

In this section, we will introduce the details of the base network. We use the one-stage method YOLACT (Bolya et al. 2019) as our base network for the reason that the prior boxes in YOLACT are agnostic to object categories. In YOLACT, the complex instance segmentation task is split into two parallel and straightforward tasks. One branch uses FCN (Long, Shelhamer, and Darrell 2015) to generate a set of image-sized "prototype masks" which does not depend on anyone instance. Another branch utilizes an object detection module to predict a vector of "mask coefficients" for each anchor. The "mask coefficients" are used to encode the representation of instances in the prototype space. Then the predicted instances are selected by NMS operation. Finally, a mask is constructed for each predicted instance by linearly combining the work of these two branches.

The generation of prototypes is similar to standard semantic segmentation, however, it differs in that there is no explic-

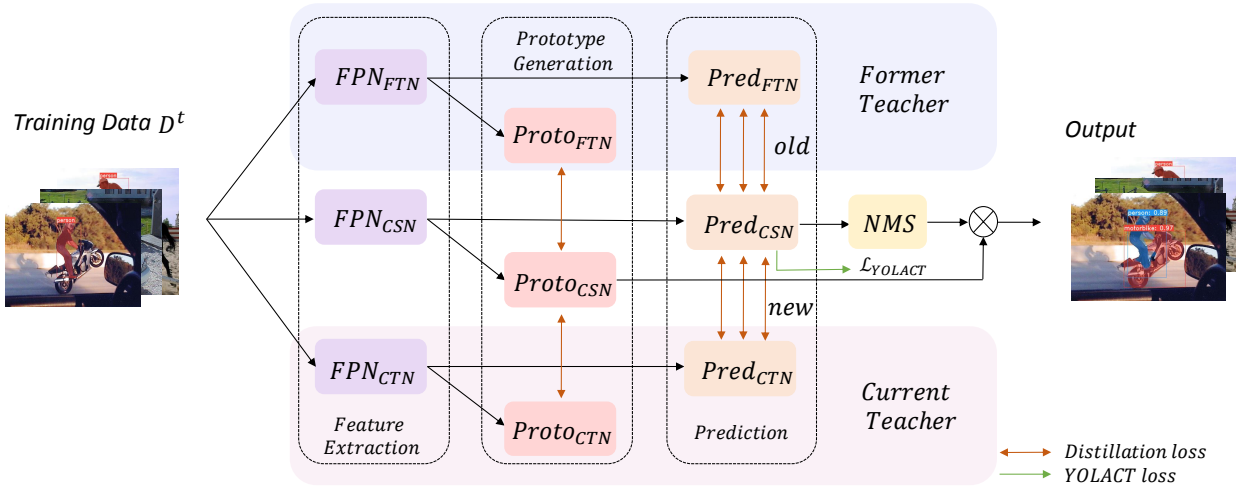


Figure 3: This is the flow of our method in task $t, t > 1$. All the networks extract the features using the Feature Pyramid Network (FPN) (Lin et al. 2017) from the input data. The extracted features will be utilized to generate prototype masks and be sent to the prediction network to produce three values: bounding box coordinates, classification scores, and mask coefficients. To preserve the old knowledge and overcome negative transfer, we compute the distillation loss between CSN and two teacher networks. The orange lines between $Proto_{FTN}$, $Proto_{CSN}$ and $Proto_{CSN}$ are the prototype distillation losses. The orange lines between $Pred_{FTN}$, $Pred_{CTN}$ and $Pred_{CSN}$ are bounding box, classification, and mask distillation losses, respectively. The outputs of CSN are handled by NMS to produce the final output.

it loss of prototypes. The supervision of this generation process comes from the final mask loss after assembly. The prediction of the mask coefficients is achieved by adding a third branch to typical anchor-based object detectors. To subtract out prototypes from the final mask, we sent the mask coefficients to a $tanh$ layer to produce more stable outputs.

Three losses are used to train the network: classification loss $\mathcal{L}_{cls}$, box regression loss $\mathcal{L}_{bbx}$, and mask loss $\mathcal{L}_{mask}$ with weights 1, 1.5, and 6.125 respectively. The classification loss and box regression loss are the same as (Liu et al. 2016). And the mask loss is computed by the pixel-wise binary cross entropy:

$$\mathcal{L}_{mask} = BCE(M, M_{GT}), \tag{1}$$

where M is the predicted mask and M_{GT} is the ground truth mask. The total loss is formulated as:

$$\mathcal{L}_{YOLACT} = \mathcal{L}_{cls} + 1.5\mathcal{L}_{bbx} + 6.125\mathcal{L}_{mask}. \tag{2}$$

Multi-Teacher Networks

The flow of our method is shown in Figure 2. Our method consists of three main parts in task t ($t > 1$): Former Teacher Network (FTN), Current Student Network (CSN), and Current Teacher Network (CTN).

For task t, FTN is a model pre-trained on $D^{t-1} = \{(x, y)|y \in \mathcal{C}^{t-1}\}$, it can segment instances belonging to class $\mathcal{C}^i, i <= t-1$. CSN is built upon FTN, which is obtained by increasing a prediction layer for the new classes. And CSN uses the parameters of the pre-trained FTN as its initialization parameters. The initialization of the new prediction layers is accomplished by CTN, then the initialized CSN is trained on $D^t = \{(x, y)|y \in \mathcal{C}^t\}$. The goal of CSN is to segment instances belonging to class $\mathcal{C}^i, i <= t$. In order to achieve this goal without accessing to D^{t-1}, it needs to accept the supervision of FTN and CTN. CTN is also trained on $D^t = \{(x, y)|y \in \mathcal{C}^t\}$, different from CSN, the goal of CTN is to only segment instances belonging to class $\mathcal{C}^i, i = t$. This network does not consider the information from the previous task. Therefore, it can achieve better performance on the current task. CTN can also be called Expert Network, which means that this network specializes in segmenting instances belonging to the new classes. It will give CSN more accurate guidance on new classes to alleviate the influence of negative transfer.

To overcome catastrophic forgetting and negative transfer problems in incremental instance segmentation, as shown in Figure 3, we compute the distillation losses between CSN, FTN and CTN. Concretely, to preserve the previous knowledge, we compute the distillation losses between CSN and FTN. To help CSN better adapt to new classes, we also compute distillation losses between CSN and CTN besides the standard YOLACT loss function.

The purpose of distillation loss is that the two networks participating in the distillation maintain the same output for the same input. In the compute of proposed distillation losses, the parameters of the two teacher networks are frozen. It means that CSN needs to make its output close to its two teachers.

Given an image I^t in $D^t = \{(x, y)|y \in \mathcal{C}^t\}$, the responses of the FTN are recorded as $FTN(I^t)$, consist-

ing of instance prototype masks FTN_{proto}, mask coefficients FTN_{mask}, class logits FTN_{cls}, and bounding box regressions FTN_{bbx}. Even if no instance is segmented by FTN, the $FTN(I^t)$ can carry enough information about the previous classes to $CSN(I^t)$ by distillation. Specially, the class logits for the new classes C^t are not considered in the loss between $FTN(I^t)$ and $CSN(I^t)$. The definition of $CSN(I^t)$ is similar to $FTN(I^t)$. Similarly, the responses of CTN are recorded as $CTN(I^t)$, including $CTN_{proto}, CTN_{mask}, CTN_{cls}$, and CTN_{bbx}. The class logits for the old classes $\{C^1, \ldots, C^{t-1}\}$ are not considered in the loss between $CTN(I^t)$ and $CSN(I^t)$. $CTN(I^t)$ will give a good initialization and soft supervision to the learning of $CSN(I^t)$, which indeed reduces the influence of negative transfer. We make a lot of ablation experiments to validate the effectiveness of multi-teacher learning, which will be shown in the experimental part. We use $L2$ loss to compute the distillation losses. Therefore, the loss formulations between CSN and two teacher networks are defined as:

$$\begin{aligned}\mathcal{L}_{FTN} = [&\lambda_1(FTN_{proto} - CSN_{proto})^2 \\ &+ \lambda_1(FTN_{mask} - CSN_{mask})^2 \\ &+ \lambda_2(FTN_{cls} - CSN_{cls})^2 \\ &+ \lambda_3(FTN_{bbx} - CSN_{bbx})^2],\end{aligned} \tag{3}$$

$$\begin{aligned}\mathcal{L}_{CTN} = [&\lambda_1(CTN_{proto} - CSN_{proto})^2 \\ &+ \lambda_1(CTN_{mask} - CSN_{mask})^2 \\ &+ \lambda_2(CTN_{cls} - CSN_{cls})^2 \\ &+ \lambda_3(CTN_{bbx} - CSN_{bbx})^2].\end{aligned} \tag{4}$$

We define the sum of prototype and mask losses as instance loss. $\lambda_1, \lambda_2, \lambda_3$ are the hyper-parameters to balance the three losses.

The total loss can be formulated as:

$$\mathcal{L} = \beta_1\mathcal{L}_{FTN} + \beta_2\mathcal{L}_{YOLACT} + \beta_3\mathcal{L}_{CTN}, \tag{5}$$

where $\beta_1, \beta_2, \beta_3$ are the hyper-parameters. By adjusting the values of these three hyper-parameters, we can control the influence of catastrophic forgetting and negative transfer problem to achieve higher segmentation precision.

Experiments

Datasets and Evaluation

We evaluate our method on the PASCAL 2012 SBD (Hariharan et al. 2011) dataset and Microsoft COCO (Lin et al. 2014) dataset. PASCAL 2012 SBD has 20 object categories. It consists of 8498 training images and 2857 testing images. On the other hand, COCO has 80k images in the training set and 40k images in the validation set. The total number of object classes in COCO is 80, which includes all the classes in PASCAL 2012 SBD. We report mean average precision (mAP) weighted across different IoU from 0.5 to 0.95 on the two datasets, which can better demonstrate the overall degree of forgetting and adaption.

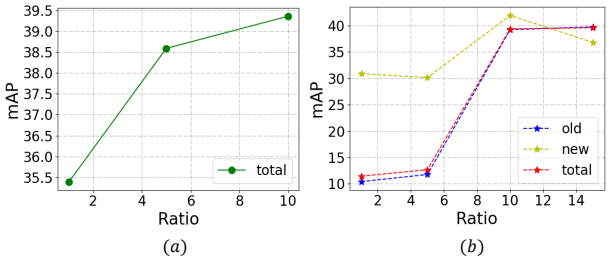


Figure 4: The influence of λ (a) and β (b) in distillation losses. The abscissa is the ratio of λ or β. For example, if the value of abscissa in (a) is x, it means that $\lambda_1 : \lambda_2 : \lambda_3$ is $x : 1 : 1$. The ordinate is the mAP on addition of one class "tvmonitor" task.

Implementation Details

We use SGD (Bottou 2012) to train all the models in the experiments. The backbone in YOLACT is ResNet-50 (He et al. 2016) and the base image size is 550×550. For the first task, we set the learning rate to 1e-4, and for the next tasks, we set the learning rate to 5e-5. For PASCAL 2012 SBD dataset, we train the model for 120k iterations, and the learning rate will decay at iterations 60k and 100k using a weight decay of 5e-4. For COCO dataset, the model is trained for 400k iterations, and the learning rate decays at iterations 140k, 300k, and 350k. The weight decay is the same as PASCAL 2012 SBD.

Addition of One Class

In the first task, we train the model using 19 classes in PASCAL 2012 SBD and then train the model on the remaining class without access to the data of previous classes in the second task. A summary of the evaluation of these networks is shown in Table 2, with the full results in Table 1.

Fine-tuning is a baseline method for adding a new class, which trains the model directly on new data without adding any constraints on the model to preserve the previous knowledge. This operation causes a rapid drop in the performance of the old classes. Besides, there are not many iterations of the fine-tuning method for the reason to keep the performance of previous classes. Therefore, it learns badly on the new class due to the lack of optimization times. As shown in Table 2, it has forgotten more than half of the previous knowledge. And the mAP of the new class only achieves 8.49%. Compared with fine-tuning, our model performs better under the guidance of two teacher networks, both in new and old classes. It has achieved 39.65% mAP on old classes and 41.29% mAP on the new class.

We also make ablation studies to validate the effectiveness of the component in our method. FTN is the most important part to preserve the performance of our method on old classes. Thus, if FTN is removed, the incomplete model will suffer from catastrophic forgetting on old classes. It is similar to fine-tuning at this point. Therefore, we train the model without FTN for the same iterations as fine-tuning. As is shown in Table 2, the mAP on old classes dropped from 39.65%

Method	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	
	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	Tv	mAP
(1-19)	53.93	36.68	45.77	27.05	28.42	68.47	50.37	66.08	19.19	36.36	
	13.14	63.29	45.35	43.37	38.88	22.03	42.59	30.40	63.25	-	41.82
+(20) fine-tuning	13.86	11.51	17.75	3.34	6.86	13.96	13.98	39.73	3.32	13.32	
	0.00	27.49	17.29	6.12	3.73	4.57	9.18	4.64	17.29	8.49	11.82
+(20) w/o FTN	4.03	0.05	1.97	1.04	0.10	5.50	1.97	0.30	0.12	0.55	
	0.18	1.69	0.16	0.18	0.05	0.35	2.14	0.46	4.83	9.74	1.77
+(20) w/o CTN	53.24	34.13	44.35	24.86	27.39	66.47	45.73	64.51	17.74	36.58	
	12.56	61.38	44.23	40.39	36.70	20.69	39.21	29.06	59.03	33.38	39.58
+(20) w/o Ins	0.02	0.00	0.00	0.32	0.40	2.03	0.01	0.00	0.00	0.00	
	0.00	0.00	0.00	0.00	0.00	0.28	0.00	0.00	0.00	34.62	1.88
+(20) w/o Bbox	52.92	33.3	43.46	25.03	26.54	65.27	44.55	63.49	16.72	34.94	
	13.65	60.55	43.39	39.76	34.09	20.07	38.77	30.71	59.90	42.86	39.50
+(20) w/o Cls	52.61	33.42	45.53	25.37	26.95	65.33	45.17	63.91	16.54	33.86	
	14.83	59.69	42.03	39.18	34.20	20.61	37.84	31.07	60.75	38.83	39.39
+(20) w two teachers	52.84	33.97	44.04	25.67	26.99	67.28	44.87	63.4	16.72	36.29	
	12.98	60.98	43.17	39.79	34.61	19.37	39.01	30.00	61.31	41.29	**39.73**
(1-20)	53.63	35.46	45.88	28.22	28.43	67.45	49.73	64.56	20.14	37.53	
	13.95	63.11	42.54	43.97	38.13	22.10	41.74	31.84	62.47	51.10	42.10

Table 1: PASCAL 2012 SBD test per-class weighted mAP (%) under different settings when the "tvmonitor" class is added. "Ins", "Bbox", "Cls" represent instance distillation loss, bounding box distillation loss and classification score distillation loss, respectively. Bold text indicates the best incremental learning performance.

to 1.36%. The efficiency of previous knowledge forgetting has been accelerated compared to fine-tuning due to the existence of CTN. In order to verify the effectiveness of CTN in learning the new class, we also remove the CTN from our method. As shown in Table 2, the mAP of "tvmonitor" is dropped from 41.29% to 33.38%.

The instance distillation loss, including prototype and mask coefficients distillation loss, is most critical in our loss function. As shown in Table 2, the model removing instance distillation almost forgets all previous knowledge. This result is very similar to the final output of a model that has been fine-tuned many iterations. The bounding box and classification distillation losses also contribute to the performance of the model, while it is relatively small compared to instance distillation loss.

Addition of Multiple Classes

In this scenario, we train the network on the first ten PASCAL 2012 SBD classes, which are in alphabetical order. And in the second task, we retrain the model on the remaining ten classes. The evaluation results of these networks are shown in Table 3.

The model trained on new ten classes achieves 37.39% mAP compared to 42.10% of the baseline network trained on all the classes. The performance degradation is mainly caused by the learning of new classes. As shown in Table 3, the baseline model trained on only the first ten classes achieves 40.82% mAP on the old classes, while the retrained model with two teachers achieves 40.27% mAP. It means that little previous knowledge is forgotten. In contrast, the learning of the new classes is not satisfactory. The retrained model with two teacher networks achieves 34.50% mAP on new classes while the baseline model (1-20) achieves 41.09% mAP. However, the learning of the new classes gets worse when we remove the CTN from the network, its mAP drops from 34.50% to 32.29%. Interestingly, the mAP on the old classes is also dropped when the CTN is removed. We guess this phenomenon is because CTN can help the model adapt to the new classes quickly, which avoids the changes of some unnecessary parameters.

We also conduct the experiments on COCO dataset. The results are shown in Table 5. The initial set contains the first 40 classes, and the remaining 40 classes are used in the second task.

Sequential Addition of Multiple Classes

We evaluate the sequential learning ability of our method. In the first task, we train the first ten classes. And we add five new classes in the second stage. Finally, we add the remaining five classes. The evaluation results of all the models are shown in Table 4.

As shown in Table 4, our method successfully preserves the old knowledge in a sequential setting. And the retention rate of old knowledge is very stable. The second learning preserves 98.7% mAP of the first learning in the old classes. And the third learning keeps 98.3% mAP of the second learning in the previous classes. Although the learning of the new classes is still not good, the addition of the CTN is a great improvement compared with learning directly on the new classes. For example, CTN helps the network improve 17.8% mAP on the new classes in the third task.

We also show some visualization results. As shown in Figure 5, the old classes like bicycle, boat and car are preserved, and the new classes person and motorbike are seg-

Method	Old	New	All
(1-19)	41.82	-	-
+ (20) fine-tuning	12.00	8.49	11.82
+ (20) w/o FTN	1.35	9.74	1.77
+ (20) w/o CTN	**39.91**	33.38	39.58
+ (20) w/o Ins	0.16	34.62	1.88
+ (20) w/o Bbox.	39.32	**42.86**	39.50
+ (20) w/o Cls	39.41	41.82	39.39
+ (20) w two teachers	39.65	41.29	**39.73**
(1-20)	41.63	51.10	42.10

Table 2: PASCAL 2012 SBD test weighted mAP (%) under different settings when the "tvmonitor" class is added. Bold text indicates the best incremental learning performance.

Method	Old	New	All
(1-10)	40.82	-	-
+ (11-20) finetune	17.82	2.07	9.95
+ (11-20) w/o FTN	2.07	9.88	5.98
+ (11-20) w/o CTN	38.69	32.29	35.49
+ (11-20) w two teachers	**40.27**	**34.50**	**37.39**
(1-20)	43.1	41.09	42.10

Table 3: PASCAL 2012 SBD test weighted mAP (%) when ten classes are added at once.

Method	Old	New	All
(1-10)	40.82	-	-
+ (11-15) w/o CTN	39.80	32.26	37.29
+ (11-15) w two teachers	**40.28**	**33.04**	**37.87**
+ (15-20) w/o CTN	39.28	24.16	33.76
+ (15-20) w two teachers	**39.63**	**29.40**	**34.51**
(1-20)			42.10

Table 4: PASCAL 2012 SBD test weighted mAP (%) when five classes are added at once and add twice in total.

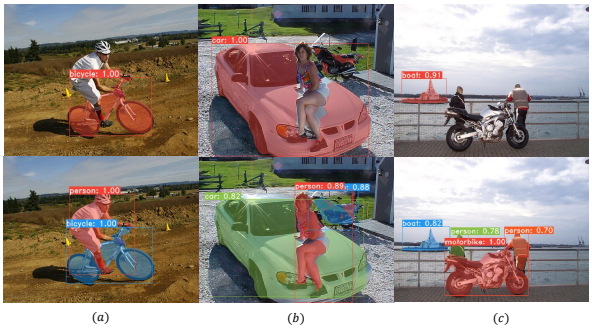
mented successfully.

The Balance of the Supervision

We adjust the value of λ and β in the distillation loss function to achieve higher mAP. The ratio of the three parameters λ_1,λ_2, and λ_3 determine the importance of three different distillation losses. And the ratio of the three parameters β_1,β_2, and β_3 decide the balance between the learning of old and new classes.

Considering the performance of the three kinds of loss in the ablation experiment, we think the value of λ_1 should be greater than λ_2 and λ_3. The experimental results also prove this point of view. As shown in Figure 4(a), the value of mAP increases as the ratio of λ_1 to λ_2, λ_3 increases. The performance growth slows down when the ratio reaches 10. Therefore, we set the ratio of $\lambda_1 : \lambda_2 : \lambda_3$ to $10 : 1 : 1$.

To seek a balance between old knowledge preservation and adaption to new classes, we adjust the value of β to change the supervision of the two teacher networks. As shown in Figure 4(b), the mAP of new classes keeps increasing until the ratio exceeds 10. And the mAPs of the old and total classes grow slowly after the ratio of 10. Thus, the ratio of $\beta_1 : \beta_2 : \beta_3$ is also set to $10 : 1 : 1$.



(a) (b) (c)

Figure 5: Visualization of some results. The color of the segmented object is random. The model first trained on the first ten classes and retrained on the latter five classes. Classes "car", "boat", and "bicycle" are in the first ten classes, while "person" and "motorbike" are in the latter five classes.

Method	mAP@.5	mAP@[.5, .95]
(1-40)+ (41-80)	31.13	16.72
(1-80)	36.62	20.82

Table 5: COCO minival (first 5000 validation images) average precision (%). We compare the model learned incrementally on half the classes with the baseline trained on all jointly.

Conclusion and Future Work

In this paper, we propose an approach for incremental learning of instance segmentation model without access to the training data of previous classes. We address the problem of catastrophic forgetting on old classes and overcome the negative transfer problem to help the model better adapt to new classes. The key to solving these two problems is to make the current task network obtain the supervision of the two teacher networks. Extensive experimental results have demonstrated that our approach performs well under different incremental settings. Our future work aims to handle the incremental instance segmentation based on two-stage methods, which requires a category-agnostic proposal network.

Acknowledgments

Our work was supported in part by the National Natural Science Foundation of China under Grant 62071361, the National Key R&D Program of China under Grant 2017YFE0104100, and the China Research Project under Grant 6141B07270429.

References

Bolya, D.; Zhou, C.; Xiao, F.; and Lee, Y. J. 2019. YOLACT: Real-time Instance Segmentation. In *CVPR*.

Bottou, L. 2012. Stochastic Gradient Descent Tricks. *Neural Networks: Tricks of the Trade: Second Edition* 421–436.

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr, P. H. S. 2018. Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Chen, H.; Sun, K.; Tian, Z.; Shen, C.; Huang, Y.; and Yan, Y. 2020. BlendMask: Top-Down Meets Bottom-Up for Instance Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chen, L.-C.; Hermans, A.; Papandreou, G.; Schroff, F.; Wang, P.; and Adam, H. 2018. MaskLab: Instance Segmentation by Refining Object Detection With Semantic and Direction Features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chen, X.; Girshick, R.; He, K.; and Dollar, P. 2019. TensorMask: A Foundation for Dense Object Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Dai, J.; He, K.; Li, Y.; Ren, S.; and Sun, J. 2016. Instance-Sensitive Fully Convolutional Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Dang, Z.; Deng, C.; Yang, X.; and Huang, H. 2020. Multi-Scale Fusion Subspace Clustering Using Similarity Constraint. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6658–6667.

Deng, C.; Yang, X.; Nie, F.; and Tao, D. 2019. Saliency Detection via a Multiple Self-Weighted Graph-Based Manifold Ranking. *IEEE Transactions on Multimedia* 22(4): 885–896.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Kai Li; and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248–255.

Dhar, P.; Singh, R. V.; Peng, K.-C.; Wu, Z.; and Chellappa, R. 2019. Learning Without Memorizing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Feng, H.; Kong, K.; Chen, M.; Zhang, T.; Zhu, M.; and Chen, W. 2020. SHOT-VAE: Semi-supervised Deep Generative Models With Label-aware ELBO Approximations. *CoRR* abs/2011.10684. URL https://arxiv.org/abs/2011.10684.

Girshick, R. 2015. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Han, X.; Wang, S.; Su, C.; Zhang, W.; Huang, Q.; and Tian, Q. 2020. Interpretable Visual Reasoning via Probabilistic Formulation Under Natural Supervision. In *European Conference on Computer Vision (ECCV)*, 553–570. Springer.

Hariharan, B.; Arbelez, P.; Bourdev, L.; Maji, S.; and Malik, J. 2011. Semantic contours from inverse detectors. In *2011 International Conference on Computer Vision (ICCV)*, 991–998.

He, K.; Gkioxari, G.; Dollar, P.; and Girshick, R. 2017. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

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 (CVPR)*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2018. Lifelong Learning via Progressive Distillation and Retrospection. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; and Grabska-Barwinska, A. a. 2016. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114(13): 3521–3526.

Li, Y.; Qi, H.; Dai, J.; Ji, X.; and Wei, Y. 2017. Fully Convolutional Instance-Aware Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, Z.; and Hoiem, D. 2017. Learning without Forgetting. *IEEE Transactions on Pattern Analysis Machine Intelligence* 1–1.

Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature Pyramid Networks for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C. Lawrence", e. D.; Pajdla, T.; Schiele, B.; and Tuytelaars, T. 2014. Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Liu, S.; Qi, L.; Qin, H.; Shi, J.; and Jia, J. 2018. Path Aggregation Network for Instance Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, Alexander C.", e. B.; Matas, J.; Sebe, N.; and Welling, M. 2016. SSD: Single Shot MultiBox Detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully Convolutional Networks for Semantic Segmentation. In *Pro-*

ceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Oquab, M.; Bottou, L.; Laptev, I.; and Sivic, J. 2014. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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 28*.

Serra, J.; Suris, D.; Miron, M.; and Karatzoglou, A. 2018. Overcoming Catastrophic Forgetting with Hard Attention to the Task. volume 80, 4548–4557. PMLR.

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 (ICCV)*.

Wei, K.; Deng, C.; and Yang, X. 2020. Lifelong Zero-Shot Learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, 551–557.

Wei, K.; Yang, M.; Wang, H.; Deng, C.; and Liu, X. 2019. Adversarial Fine-Grained Composition Learning for Unseen Attribute-Object Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 3741–3749.

Yang, E.; Deng, C.; Li, C.; Liu, W.; Li, J.; and Tao, D. 2018. Shared predictive cross-modal deep quantization. *IEEE transactions on neural networks and learning systems* 29(11): 5292–5303.

Yang, E.; Liu, M.; Yao, D.; Cao, B.; Lian, C.; Yap, P.-T.; and Shen, D. 2020a. Deep Bayesian Hashing with Center Prior for Multi-modal Neuroimage Retrieval. *IEEE transactions on medical imaging* .

Yang, X.; Deng, C.; Liu, T.; and Tao, D. 2020b. Heterogeneous graph attention network for unsupervised multiple-target domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* .

Yang, X.; Deng, C.; Zheng, F.; Yan, J.; and Liu, W. 2019. Deep spectral clustering using dual autoencoder network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4066–4075.

Yang, Y.; Qiu, J.; Song, M.; Tao, D.; and Wang, X. 2020c. Distilling Knowledge From Graph Convolutional Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yu, L.; Twardowski, B.; Liu, X.; Herranz, L.; Wang, K.; Cheng, Y.; Jui, S.; and Weijer, J. v. d. 2020. Semantic Drift Compensation for Class-Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhao, B.; Tang, S.; Chen, D.; Bilen, H.; and Zhao, R. 2021. Continual Representation Learning for Biometric Identification. *Winter Conference on Applications of Computer Vision* .

Zhou, X.; Zhuo, J.; and Krahenbuhl, P. 2019. Bottom-Up Object Detection by Grouping Extreme and Center Points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zitnick, C. L.; Dollár, Piotr", e. D.; Pajdla, T.; Schiele, B.; and Tuytelaars, T. 2014. Edge Boxes: Locating Object Proposals from Edges. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 391–405.