Modeling the Background for Incremental Learning in Semantic Segmentation

Fabio Cermelli[1,2], Massimiliano Mancini[2,3,4], Samuel Rota Bulò[5], Elisa Ricci[3,6], Barbara Caputo[1,2]
[1]Politecnico di Torino, [2]Italian Institute of Technology, [3]Fondazione Bruno Kessler,
[4]Sapienza University of Rome, [5]Mapillary Research, [6]University of Trento
{fabio.cermelli, barbara.caputo}@polito.it, mancini@diag.uniroma1.it,
samuel@mapillary.com, eliricci@fbk.eu

Abstract

Despite their effectiveness in a wide range of tasks, deep architectures suffer from some important limitations. In particular, they are vulnerable to catastrophic forgetting, i.e. *they perform poorly when they are required to update their model as new classes are available but the original training set is not retained. This paper addresses this problem in the context of semantic segmentation. Current strategies fail on this task because they do not consider a peculiar aspect of semantic segmentation: since each training step provides annotation only for a subset of all possible classes, pixels of the background class (*i.e. *pixels that do not belong to any other classes) exhibit a semantic distribution shift. In this work we revisit classical incremental learning methods, proposing a new distillation-based framework which explicitly accounts for this shift. Furthermore, we introduce a novel strategy to initialize classifier's parameters, thus preventing biased predictions toward the background class. We demonstrate the effectiveness of our approach with an extensive evaluation on the Pascal-VOC 2012 and ADE20K datasets, significantly outperforming state of the art incremental learning methods. Code can be found at* https://github.com/fcdl94/MiB.

1. Introduction

Semantic segmentation is a fundamental problem in computer vision. In the last years, thanks to the emergence of deep neural networks and to the availability of large-scale human-annotated datasets [11, 39], the state of the art has improved significantly [20, 8, 38, 19, 37]. Current approaches are derived by extending deep architectures from image-level to pixel-level classification, taking advantage of Fully Convolutional Networks (FCNs) [20]. Over the years, semantic segmentation models based on FCNs have been improved in several ways, *e.g.* by exploiting multiscale representations [19, 37], modeling spatial dependencies and contextual cues [6, 5, 8] or considering attention models [7].

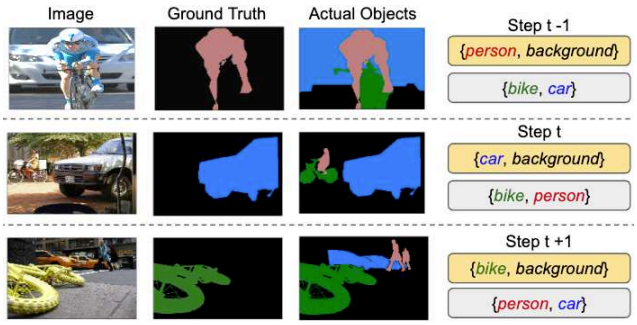

Figure 1: Illustration of the semantic shift of the background class in incremental learning for semantic segmentation. Yellow boxes denote the ground truth provided in the learning step, while grey boxes denote classes not labeled. As different learning steps have different label spaces, at step t old classes (*e.g. person*) and unseen ones (*e.g. car*) might be labeled as background in the current ground truth. Here we show the specific case of single class learning steps, but we address the general case where an arbitrary number of classes is added.

Still, existing semantic segmentation methods are not designed to incrementally update their inner classification model when new categories are discovered. While deep nets are undoubtedly powerful, it is well known that their capabilities in an incremental learning setting are limited [16]. In fact, deep architectures struggle in updating their parameters for learning new categories whilst preserving good performance on the old ones (*catastrophic forgetting* [23]).

While the problem of incremental learning has been traditionally addressed in object recognition [18, 17, 4, 28, 15] and detection [32], much less attention has been devoted to semantic segmentation. Here we fill this gap, proposing an incremental class learning (ICL) approach for semantic segmentation. Inspired by previous methods on image classification [18, 28, 3], we cope with catastrophic forgetting by resorting to knowledge distillation [14]. However, we argue (and experimentally demonstrate) that a naive application of previous knowledge distillation strategies would

not suffice in this setting. In fact, one peculiar aspect of semantic segmentation is the presence of a special class, the background class, indicating pixels not assigned to any of the given object categories. While the presence of this class marginally influences the design of traditional, offline semantic segmentation methods, this is not true in an incremental learning setting. As illustrated in Fig. 1, it is reasonable to assume that the semantics associated to the background class changes over time. In other words, pixels associated to the background during a learning step may be assigned to a specific object class in subsequent steps or vice-versa, with the effect of exacerbating the catastrophic forgetting. To overcome this issue, we revisit the classical distillation-based framework for incremental learning [18] by introducing two novel loss terms to properly account for the semantic distribution shift within the background class, thus introducing the first ICL approach tailored to semantic segmentation. We extensively evaluate our method on two datasets, Pascal-VOC [11] and ADE20K [39], showing that our approach, coupled with a novel classifier initialization strategy, largely outperform traditional ICL methods.
To summarize, the contributions of this paper are as follows:

- We study the task of incremental class learning for semantic segmentation, analyzing in particular the problem of distribution shift arising due to the presence of the background class.
- We propose a new objective function and introduce a specific classifier initialization strategy to explicitly cope with the evolving semantics of the background class. We show that our approach greatly alleviates the catastrophic forgetting, leading to the state of the art.
- We benchmark our approach over several previous ICL methods on two popular semantic segmentation datasets, considering different experimental settings. We hope that our results will serve as a reference for future works.

2. Related Works

Semantic Segmentation. Deep learning has enabled great advancements in semantic segmentation [20, 8, 38, 19, 37]. State of the art methods are based on Fully Convolutional Neural Networks [20, 2] and use different strategies to condition pixel-level annotations on their global context, *e.g.* using multiple scales [38, 19, 6, 5, 37, 8] and/or modeling spatial dependencies [6, 12]. The vast majority of semantic segmentation methods considers an offline setting, *i.e.* they assume that training data for all classes is available beforehand. To our knowledge, the problem of ICL in semantic segmentation has been addressed only in [26, 27, 33, 24]. Ozdemir *et al*. [26, 27] describe an ICL approach for medical imaging, extending a standard image-level classification method [18] to segmentation and devising a strategy to select relevant samples of old datasets for rehearsal. Taras *et al*. proposed a similar approach for segmenting remote sensing data. Differently, Michieli *et al*. [24] consider ICL for semantic segmentation in a particular setting where labels are provided for old classes while learning new ones. Moreover, they assume the novel classes to be never present as background in pixels of previous learning steps. These assumptions strongly limit the applicability of their method.

Here we propose a more principled formulation of the ICL problem in semantic segmentation. In contrast with previous works, we do not limit our analysis to medical [26] or remote sensing data [33] and we do not impose any restrictions on how the label space should change across different learning steps [24]. Moreover, we are the first to provide a comprehensive experimental evaluation of state of the art ICL methods on commonly used semantic segmentation benchmarks and to explicitly introduce and tackle the semantic shift of the background class, a problem recognized but largely overseen by previous works [24].

Incremental Learning. The problem of catastrophic forgetting [23] has been extensively studied for image classification tasks [9]. Previous works can be grouped in three categories [9]: replay-based [28, 3, 31, 15, 34, 25], regularization-based [17, 4, 36, 18, 10], and parameter isolation-based [22, 21, 30]. In replay-based methods, examples of previous tasks are either stored [28, 3, 15, 35] or generated [31, 34, 25] and then replayed while learning the new task. Parameter isolation-based methods [22, 21, 30] assign a subset of the parameters to each task to prevent forgetting. Regularization-based methods can be divided in prior-focused and data-focused. The former [36, 4, 17, 1] define knowledge as the parameters value, constraining the learning of new tasks by penalizing changes of important parameters for old ones. The latter [18, 10] exploit distillation [14] and use the distance between the activations produced by the old network and the new one as a regularization term to prevent catastrophic forgetting.

Despite these progresses, very few works have gone beyond image-level classification. A first work in this direction is [32] which considers ICL in object detection proposing a distillation-based method adapted from [18] for tackling novel class recognition and bounding box proposals generation. In this work we also take a similar approach to [32] and we resort on distillation. However, here we propose to address the problem of modeling the background shift which is peculiar of the semantic segmentation setting.

3. Method

3.1. Problem Definition and Notation

Before delving into the details of ICL for semantic segmentation, we first introduce the task of semantic segmen-

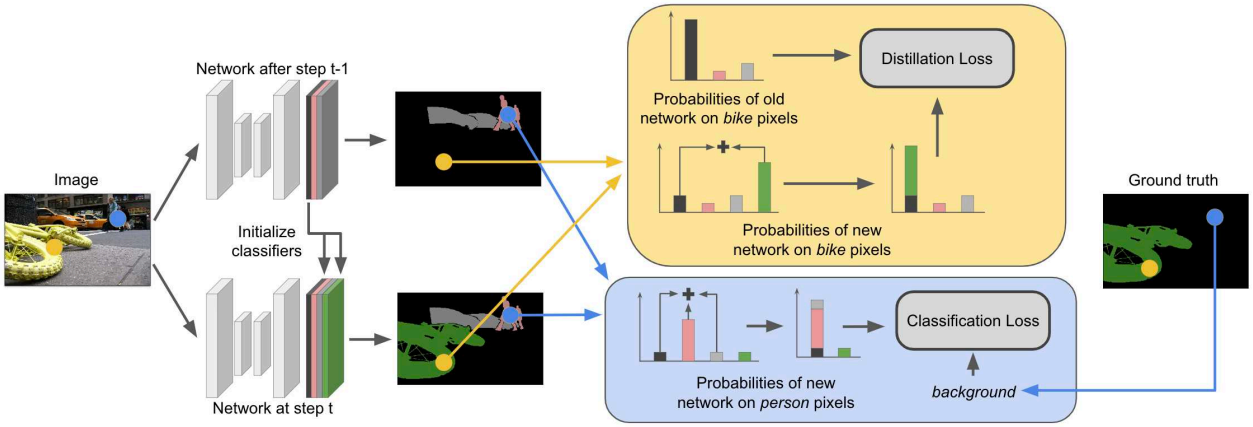

Figure 2: Overview of our method. At learning step t an image is processed by the old (top) and current (bottom) models, mapping the image to their respective output spaces. As in standard ICL methods, we apply a cross-entropy loss to learn new classes (blue block) and a distillation loss to preserve old knowledge (yellow block). In this framework, we model the semantic changes of the background across different learning steps by (i) initializing the new classifier using the weights of the old background one (left), (ii) comparing the pixel-level background ground truth in the cross-entropy with the probability of having either the background (black) or an old class (pink and grey bars) and (iii) relating the background probability given by the old model in the distillation loss with the probability of having either the background or a novel class (green bar).

tation. Let us denote as $\mathcal{X}$ the input space (*i.e.* the image space) and, without loss of generality, let us assume that each image $x \in \mathcal{X}$ is composed by a set of pixels $\mathcal{I}$ with constant cardinality $|\mathcal{I}| = N$. The output space is defined as $\mathcal{Y}^N$, with the latter denoting the product set of N-tuples with elements in a label space $\mathcal{Y}$. Given an image x the goal of semantic segmentation is to assign each pixel x_i of image x a label $y_i \in \mathcal{Y}$, representing its semantic class. Out-of-class pixels can be assigned a special class, *i.e.* the background class $\mathtt{b} \in \mathcal{Y}$. Given a training set $\mathcal{T} \subset \mathcal{X} \times \mathcal{Y}^N$, the mapping is realized by learning a model f_θ with parameters θ from the image space $\mathcal{X}$ to a pixel-wise class probability vector, *i.e.* $f_\theta : \mathcal{X} \mapsto \mathbb{R}^{N \times |\mathcal{Y}|}$. The output segmentation mask is obtained as $y^* = \{\arg\max_{c \in \mathcal{Y}} f_\theta(x)[i,c]\}_{i=1}^N$, where $f_\theta(x)[i,c]$ is the probability for class c in pixel i.

In the ICL setting, training is realized over multiple phases, called *learning steps*, and each step introduces novel categories to be learnt. In other terms, during the t_{th} learning step, the previous label set $\mathcal{Y}^{t-1}$ is expanded with a set of new classes $\mathcal{C}^t$, yielding a new label set $\mathcal{Y}^t = \mathcal{Y}^{t-1} \cup \mathcal{C}^t$. At learning step t we are also provided with a training set $\mathcal{T}^t \subset \mathcal{X} \times (\mathcal{C}^t)^N$ that is used in conjunction to the previous model $f_{\theta^{t-1}} : \mathcal{X} \mapsto \mathbb{R}^{N \times |\mathcal{Y}^{t-1}|}$ to train an updated model $f_{\theta^t} : \mathcal{X} \mapsto \mathbb{R}^{N \times |\mathcal{Y}^t|}$. As in standard ICL, in this paper we assume the sets of labels $\mathcal{C}^t$ that we obtain at the different learning steps to be disjoint, except for the special void/background class $\mathtt{b}$.

3.2. Incremental Learning for Semantic Segmentation with Background Modeling

A naive approach to address the ICL problem consists in retraining the model f_{θ^t} on each set $\mathcal{T}^t$ sequentially. When the predictor f_{θ^t} is realized through a deep architecture, this corresponds to fine-tuning the network parameters on the training set $\mathcal{T}^t$ initialized with the parameters θ^{t-1} from the previous stage. This approach is simple, but it leads to catastrophic forgetting. Indeed, when training using $\mathcal{T}^t$ no samples from the previously seen object classes are provided. This biases the new predictor f_{θ^t} towards the novel set of categories in $\mathcal{C}^t$ to the detriment of the classes from the previous sets. In the context of ICL for image-level classification, a standard way to address this issue is coupling the supervised loss on $\mathcal{T}^t$ with a regularization term, either taking into account the importance of each parameter for previous tasks [17, 31], or by distilling the knowledge using the predictions of the old model $f_{\theta^{t-1}}$ [18, 28, 3]. We take inspiration from the latter solution to initialize the overall objective function of our problem. In particular, we minimize a loss function of the form:

$$\mathcal{L}(\theta^t) = \frac{1}{|\mathcal{T}^t|} \sum_{(x,y) \in \mathcal{T}^t} \left(\ell_{ce}^{\theta^t}(x,y) + \lambda \ell_{kd}^{\theta^t}(x) \right) \qquad (1)$$

where ℓ_{ce} is a standard supervised loss (*e.g.* cross-entropy loss), ℓ_{kd} is the distillation loss and $\lambda > 0$ is a hyper-parameter balancing the importance of the two terms.

As stated in Sec. 3.1, differently from standard ICL settings considered for image classification problems, in semantic segmentation we have that two different label sets $\mathcal{C}^s$ and $\mathcal{C}^u$ share the common void/background class $\mathtt{b}$. However, the distribution of the background class changes across different incremental steps. In fact, background annotations given in $\mathcal{T}^t$ refer to classes not present in $\mathcal{C}^t$, that might belong to the set of seen classes $\mathcal{Y}^{t-1}$ and/or to still unseen classes *i.e.* $\mathcal{C}^u$ with $u > t$ (see Fig. 1). In the following, we

show how we account for the semantic shift in the distribution of the background class by revisiting standard choices for the general objective defined in Eq. (1).

Revisiting Cross-Entropy Loss. In Eq.(1), a possible choice for ℓ_{ce} is the standard cross-entropy loss computed over all image pixels:

$$\ell_{ce}^{\theta^t}(x,y) = -\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \log q_x^t(i,y_i)\,, \tag{2}$$

where $y_i \in \mathcal{Y}^t$ is the ground truth label associated to pixel i and $q_x^t(i,c) = f_{\theta^t}(x)[i,c]$.

The problem with Eq.(2) is that the training set $\mathcal{T}^t$ we use to update the model only contains information about novel classes in $\mathcal{C}^t$. However, the background class in $\mathcal{T}^t$ might include also pixels associated to the previously seen classes in $\mathcal{Y}^{t-1}$. In this paper, we argue that, without explicitly taking into account this aspect, the catastrophic forgetting problem would be even more severe. In fact, we would drive our model to predict the background label $\mathtt{b}$ for pixels of old classes, further degrading the capability of the model to preserve semantic knowledge of past categories. To avoid this issue, in this paper we propose to modify the cross-entropy loss in Eq.(2) as follows:

$$\ell_{ce}^{\theta^t}(x,y) = -\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \log \tilde{q}_x^t(i,y_i)\,, \tag{3}$$

where:

$$\tilde{q}_x^t(i,c) = \begin{cases} q_x^t(i,c) & \text{if } c \neq \mathtt{b} \\ \sum_{k\in\mathcal{Y}^{t-1}} q_x^t(i,k) & \text{if } c = \mathtt{b}\,. \end{cases} \tag{4}$$

Our intuition is that by using Eq.(3) we can update the model to predict the new classes and, at the same time, account for the uncertainty over the actual content of the background class. In fact, in Eq.(3) the background class ground truth is not directly compared with its probabilities $q_x^t(i,\mathtt{b})$ obtained from the current model f_{θ^t}, but with the probability of having *either an old class or the background*, as predicted by f_{θ^t} (Eq.(4)). A schematic representation of this procedure is depicted in Fig. 2 (blue block). It is worth noting that the alternative of ignoring the background pixels within the cross-entropy loss is a sub-optimal solution. In fact, this would not allow to adapt the background classifier to its semantic shift and to exploit the information that new images might contain about old classes.

Revisiting Distillation Loss. In the context of incremental learning, distillation loss [14] is a common strategy to transfer knowledge from the old model $f_{\theta^{t-1}}$ into the new one, preventing catastrophic forgetting. Formally, a standard choice for the distillation loss ℓ_{kd} is:

$$\ell_{kd}^{\theta^t}(x,y) = -\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\sum_{c\in\mathcal{Y}^{t-1}} q_x^{t-1}(i,c)\log \hat{q}_x^t(i,c)\,, \tag{5}$$

where $\hat{q}_x^t(i,c)$ is defined as the probability of class c for pixel i given by f_{θ^t} but re-normalized across all the classes in $\mathcal{Y}^{t-1}$ *i.e.*:

$$\hat{q}_x^t(i,c) = \begin{cases} 0 & \text{if } c \in \mathcal{C}^t \setminus \{\mathtt{b}\} \\ q_x^t(i,c)/\sum_{k\in\mathcal{Y}^{t-1}} q_x^t(i,k) & \text{if } c \in \mathcal{Y}^{t-1}\,. \end{cases} \tag{6}$$

The rationale behind ℓ_{kd} is that f_{θ^t} should produce activations close to the ones produced by $f_{\theta^{t-1}}$. This regularizes the training procedure in such a way that the parameters θ^t are still anchored to the solution found for recognizing pixels of the previous classes, *i.e.* θ^{t-1}.

The loss defined in Eq.(5) has been used either in its base form or variants in different contexts, from incremental task [18] and class learning [28, 3] in object classification to complex scenarios such as detection [32] and segmentation [24]. Despite its success, it has a fundamental drawback in semantic segmentation: it completely ignores the fact that the background class is shared among different learning steps. While with Eq.(3) we tackled the first problem linked to the semantic shift of the background (*i.e.* $\mathtt{b} \in \mathcal{T}^t$ contains pixels of $\mathcal{Y}^{t-1}$), we use the distillation loss to tackle the second: annotations for background in $\mathcal{T}^s$ with $s < t$ might include pixels of classes in $\mathcal{C}^t$.

From the latter considerations, the background probabilities assigned to a pixel by the old predictor $f_{\theta^{t-1}}$ and by the current model f_{θ^t} do not share the same semantic content. More importantly, $f_{\theta^{t-1}}$ might predict as background pixels of classes in $\mathcal{C}^t$ that we are currently trying to learn. Notice that this aspect is peculiar to the segmentation task and it is not considered in previous incremental learning models. However, in our setting we must explicitly take it into account to perform a correct distillation of the old model into the new one. To this extent we define our novel distillation loss by rewriting $\hat{q}_x^t(i,c)$ in Eq.(6) as:

$$\hat{q}_x^t(i,c) = \begin{cases} q_x^t(i,c) & \text{if } c \neq \mathtt{b} \\ \sum_{k\in\mathcal{C}^t} q_x^t(i,k) & \text{if } c = \mathtt{b}\,. \end{cases} \tag{7}$$

Similarly to Eq.(5), we still compare the probability of a pixel belonging to seen classes assigned by the old model, with its counterpart computed with the current parameters θ^t. However, differently from classical distillation, in Eq.(7) the probabilities obtained with the current model are kept unaltered, *i.e.* normalized across the whole label space $\mathcal{Y}^t$ and not with respect to the subset $\mathcal{Y}^{t-1}$ (Eq.(6)). More importantly, the background class probability as given by $f_{\theta^{t-1}}$ is not directly compared with its counterpart in f_{θ^t}, but with the probability of having *either a new class or the background*, as predicted by f_{θ^t} (see Fig. 2, yellow block).

We highlight that, with respect to Eq.(6) and other simple choices (*e.g.* excluding $\mathtt{b}$ from Eq.(6)) this solution has two advantages. First, we can still use the full output space of

the old model to distill knowledge in the current one, without any constraint on pixels and classes. Second, we can propagate the uncertainty we have on the semantic content of the background in $f_{\theta^{t-1}}$ without penalizing the probabilities of new classes we are learning in the current step t.

Classifiers' Parameters Initialization. As discussed above, the background class $\mathtt{b}$ is a special class devoted to collect the probability that a pixel belongs to an unknown object class. In practice, at each learning step t, the novel categories in $\mathcal{C}^t$ are unknowns for the old classifier $f_{\theta^{t-1}}$. As a consequence, unless the appearance of a class in $\mathcal{C}^t$ is very similar to one in $\mathcal{Y}^{t-1}$, it is reasonable to assume that $f_{\theta^{t-1}}$ will likely assign pixels of $\mathcal{C}^t$ to $\mathtt{b}$. Taking into account this initial bias on the predictions of f_{θ^t} on pixels of $\mathcal{C}^t$, it is detrimental to randomly initialize the classifiers for the novel classes. In fact a random initialization would provoke a misalignment among the features extracted by the model (aligned with the background classifier) and the random parameters of the classifier itself. Notice that this could lead to possible training instabilities while learning novel classes since the network could initially assign high probabilities for pixels in $\mathcal{C}^t$ to $\mathtt{b}$.

To address this issue, we propose to initialize the classifier's parameters for the novel classes in such a way that given an image x and a pixel i, the probability of the background $q_x^{t-1}(i,\mathtt{b})$ is uniformly spread among the classes in $\mathcal{C}^t$, *i.e.* $q_x^t(i,c) = q_x^{t-1}(i,\mathtt{b})/|\mathcal{C}^t| \;\forall c \in \mathcal{C}^t$, where $|\mathcal{C}^t|$ is the number of new classes (notice that $\mathtt{b} \in \mathcal{C}^t$). To this extent, let us consider a standard fully connected classifier and let us denote as $\{\omega_c^t, \beta_c^t\} \in \theta^t$ the classifier parameters for a class c at learning step t, with ω and β denoting its weights and bias respectively. We can initialize $\{\omega_c^t, \beta_c^t\}$ as follows:

$$\omega_c^t = \begin{cases} \omega_{\mathtt{b}}^{t-1} & \text{if } c \in \mathcal{C}^t \\ \omega_c^{t-1} & \text{otherwise} \end{cases} \tag{8}$$

$$\beta_c^t = \begin{cases} \beta_{\mathtt{b}}^{t-1} - \log(|\mathcal{C}^t|) & \text{if } c \in \mathcal{C}^t \\ \beta_c^{t-1} & \text{otherwise} \end{cases} \tag{9}$$

where $\{\omega_{\mathtt{b}}^{t-1}, \beta_{\mathtt{b}}^{t-1}\}$ are the weights and bias of the background classifier at the previous learning step. The fact that the initialization defined in Eq.(8) and (9) leads to $q_x^t(i,c) = q_x^{t-1}(i,\mathtt{b})/|\mathcal{C}^t| \;\forall c \in \mathcal{C}^t$ is easy to obtain from $q_x^t(i,c) \propto \exp(\omega_{\mathtt{b}}^t \cdot x + \beta_{\mathtt{b}}^t)$.

As we will show in the experimental analysis, this simple initialization procedure brings benefits in terms of both improving the learning stability of the model and the final results, since it eases the role of the supervision imposed by Eq.(3) while learning new classes and follows the same principles used to derive our distillation loss (Eq.(7)).

4. Experiments

4.1. ICL Baselines

We compare our method against standard ICL baselines, originally designed for classification tasks, on the considered segmentation task, thus segmentation is treated as a pixel-level classification problem. Specifically, we report the results of six different regularization-based methods, three prior-focused and three data-focused approaches.

In the first category, we chose Elastic Weight Consolidation (EWC) [17], Path Integral (PI) [36], and Riemannian Walks (RW) [4]. They employ different strategies to compute the importance of each parameter for old classes: EWC uses the empirical Fisher matrix, PI uses the learning trajectory, while RW combines EWC and PI in a unique model. We choose EWC since it is a standard baseline employed also in [32] and PI and RW since they are two simple applications of the same principle. Since these methods act at the parameter level, to adapt them to the segmentation task we keep the loss in the output space unaltered (*i.e.* standard cross-entropy across the whole segmentation mask), computing the parameters' importance by considering their effect on learning old classes.

For the data-focused methods, we chose Learning without forgetting (LwF) [18], LwF multi-class (LwF-MC) [28] and the segmentation method of [24] (ILT). We denote as LwF the original distillation based objective as implemented in Eq.(1) with basic cross-entropy and distillation losses, which is the same as [18] except that distillation and cross-entropy share the same label space and classifier. LwF-MC is the single-head version of [18] as adapted from [28]. It is based on multiple binary classifiers, with the target labels defined using the ground truth for novel classes (*i.e.* $\mathcal{C}^t$) and the probabilities given by the old model for the old ones (*i.e.* $\mathcal{Y}^{t-1}$). Since the background class is both in $\mathcal{C}^t$ and $\mathcal{Y}^{t-1}$ we implement LwF-MC by a weighted combination of two binary cross-entropy losses, on both the ground truth and the probabilities given by $f_{\theta^{t-1}}$. Finally, ILT [24] is the only method specifically proposed for ICL in semantic segmentation. It uses a distillation loss in the output space, as in our adapted version of LwF [18] and/or another distillation loss in the features space, attached to the output of the network decoder. Here, we use the variant where both losses are employed. As done by [32], we do not compare with replay-based methods (*e.g.* [28]) since they violate the standard ICL assumption regarding the unavailability of old data.

In all tables we report other two baselines: simple fine-tuning (FT) on each $\mathcal{T}^t$ (*e.g.* Eq.(2)) and training on all classes offline (Joint). The latter can be regarded as an upper bound. In the tables we denote our method as MiB (**M**ode**l**ing the **B**ackground for incremental learning in semantic segmentation). All results are reported as mean

Table 1: Mean IoU on the Pascal-VOC 2012 dataset for different incremental class learning scenarios.

	19-1						15-5						15-1					
	Disjoint			Overlapped			Disjoint			Overlapped			Disjoint			Overlapped		
Method	*1-19*	*20*	*all*	*1-19*	*20*	*all*	*1-15*	*16-20*	*all*	*1-15*	*16-20*	*all*	*1-15*	*16-20*	*all*	*1-15*	*16-20*	*all*
FT	5.8	12.3	6.2	6.8	12.9	7.1	1.1	33.6	9.2	2.1	33.1	9.8	0.2	1.8	0.6	0.2	1.8	0.6
PI [36]	5.4	14.1	5.9	7.5	14.0	7.8	1.3	34.1	9.5	1.6	33.3	9.5	0.0	1.8	0.4	0.0	1.8	0.5
EWC [17]	23.2	16.0	22.9	26.9	14.0	26.3	26.7	37.7	29.4	24.3	35.5	27.1	0.3	4.3	1.3	0.3	4.3	1.3
RW [4]	19.4	15.7	19.2	23.3	14.2	22.9	17.9	36.9	22.7	16.6	34.9	21.2	0.2	5.4	1.5	0.0	5.2	1.3
LwF [18]	53.0	9.1	50.8	51.2	8.5	49.1	58.4	37.4	53.1	58.9	36.6	53.3	0.8	3.6	1.5	1.0	3.9	1.8
LwF-MC [28]	63.0	13.2	60.5	64.4	13.3	61.9	67.2	41.2	60.7	58.1	35.0	52.3	4.5	7.0	5.2	6.4	8.4	6.9
ILT [24]	69.1	16.4	66.4	67.1	12.3	64.4	63.2	39.5	57.3	66.3	40.6	59.9	3.7	5.7	4.2	4.9	7.8	5.7
MiB	**69.6**	**25.6**	**67.4**	**70.2**	**22.1**	**67.8**	**71.8**	**43.3**	**64.7**	**75.5**	**49.4**	**69.0**	**46.2**	**12.9**	**37.9**	**35.1**	**13.5**	**29.7**
Joint	77.4	78.0	77.4	77.4	78.0	77.4	79.1	72.6	77.4	79.1	72.6	77.4	79.1	72.6	77.4	79.1	72.6	77.4

Intersection-over-Union (mIoU) in percentage, averaged over all the classes of a learning step and all the steps.

4.2. Implementation Details

For all methods we use the Deeplab-v3 architecture [6] with a ResNet-101 [13] backbone and output stride of 16. Since memory requirements are an important issue in semantic segmentation, we use in-place activated batch normalization, as proposed in [29]. The backbone has been initialized using the ImageNet pretrained model [29]. We follow [6], training the network with SGD and the same learning rate policy, momentum and weight decay. We use an initial learning rate of 10^{-2} for the first learning step and 10^{-3} for the followings, as in [32]. We train the model with a batch-size of 24 for 30 epochs for Pascal-VOC 2012 and 60 epochs for ADE20K in every learning step. We apply the same data augmentation of [6] and we crop the images to 512×512 during both training and test. For setting the hyper-parameters of each method, we use the protocol of incremental learning defined in [9], using 20% of the training set as validation. The final results are reported on the standard validation set of the datasets.

4.3. Pascal-VOC 2012

PASCAL-VOC 2012 [11] is a widely used benchmark that includes 20 foreground object classes. Following [24, 32], we define two experimental settings, depending on how we sample images to build the incremental datasets. Following [24], we define an experimental protocol called the *disjoint* setup: each learning step contains a unique set of images, whose pixels belong to classes seen either in the current or in the previous learning steps. Differently from [24], at each step we assume to have only labels for pixels of novel classes, while the old ones are labeled as background in the ground truth. The second setup, that we denote as *overlapped*, follows what done in [32] for detection: each training step contains all the images that have at least one pixel of a novel class, with only the latter annotated. It is important to note a difference with respect to the previous setup: images may now contain pixels of classes that we will learn in the future, but labeled as background. This is a more realistic setup since it does not make any assumption on the objects present in the images.

As done by previous works [32, 24], we perform three different experiments concerning the addition of one class (*19-1*), five classes all at once (*15-5*), and five classes sequentially (*15-1*), following the alphabetical order of the classes to split the content of each learning step.

Addition of one class *(19-1)*. In this experiment, we perform two learning steps: the first in which we observe the first 19 classes, and the second where we learn the *tv-monitor* class. Results are reported in Table 1. Without employing any regularization strategy, the performance on past classes drops significantly. FT, in fact, performs poorly, completely forgetting the first 19 classes. Unexpectedly, using PI as a regularization strategy does not provide benefits, while EWC and RW improve performance of nearly 15%. However, prior-focused strategies are not competitive with data-focused ones. In fact, LwF, LwF-MC, and ILT, outperform them by a large margin, confirming the effectiveness of this approch on preventing catastrophic forgetting. While ILT surpasses standard ICL baselines, our model is able to obtain a further boost. This improvement is remarkable for new classes, where we gain 11% in mIoU, while do not experience forgetting on old classes. It is especially interesting to compare our method with the baseline LwF which uses the same principles of ours but without modeling the background. Compared to LwF we achieve an average improvement of about 15%, thus demonstrating the importance of modeling the background in ICL for semantic segmentation. These results are consistent in both the *disjoint* and *overlapped* scenarios.

Single-step addition of five classes (*15-5*). In this setting we add, after the first training set, the following classes: *plant, sheep, sofa, train, tv-monitor*. Results are reported in Table 1. Overall, the behavior on the first 15 classes is consistent with the 19-1 setting: FT and PI suffer a large performance drop, data-focused strategies (LwF, LwF-MC, ILT) outperform EWC and RW by far, while our method gets the

Table 2: Mean IoU on the ADE20K dataset for different incremental class learning scenarios.

	100-50			100-10							50-50			
Method	*1-100*	*101-150*	*all*	*1-100*	*100-110*	*110-120*	*120-130*	*130-140*	*140-150*	*all*	*1-50*	*51-100*	*101-150*	*all*
FT	0.0	24.9	8.3	0.0	0.0	0.0	0.0	0.0	16.6	1.1	0.0	0.0	22.0	7.3
LwF [18]	21.1	25.6	22.6	0.1	0.0	0.4	2.6	4.6	16.9	1.7	5.7	12.9	22.8	13.9
LwF-MC [28]	34.2	10.5	26.3	18.7	2.5	8.7	4.1	6.5	5.1	14.3	27.8	7.0	10.4	15.1
ILT [24]	22.9	18.9	21.6	0.3	0.0	1.0	2.1	4.6	10.7	1.4	8.4	9.7	14.3	10.8
MiB	**37.9**	**27.9**	**34.6**	**31.8**	**10.4**	**14.8**	**12.8**	**13.6**	**18.7**	**25.9**	**35.5**	**22.2**	**23.6**	**27.0**
Joint	44.3	28.2	38.9	44.3	26.1	42.8	26.7	28.1	17.3	38.9	51.1	38.3	28.2	38.9

Table 3: Ablation study of the proposed method on the Pascal-VOC 2012 *overlapped* setup. *CE* and *KD* denote our cross-entropy and distillation losses, while *init* our initialization strategy.

	19-1			15-5			15-1		
	1-19	*20*	*all*	*1-15*	*16-20*	*all*	*1-15*	*16-20*	*all*
LwF [18]	51.2	8.5	49.1	58.9	36.6	53.3	1.0	3.9	1.8
+ *CE*	57.6	9.9	55.2	63.2	38.1	57.0	12.0	3.7	9.9
+ *KD*	66.0	11.9	63.3	72.9	46.3	66.3	34.8	4.5	27.2
+ *init*	**70.2**	**22.1**	**67.8**	**75.5**	**49.4**	**69.0**	**35.1**	**13.5**	**29.7**

best results, obtaining performances closer to the joint training upper bound. For what concerns the *disjoint* scenario, our method improves over the best baseline of 4.6% on old classes, of 2% on novel ones and of 4% in all classes. These gaps increase in the *overlapped* setting where our method surpasses the baselines by nearly 10% in all cases, clearly demonstrating its ability to take advantage of the information contained in the background class.

Multi-step addition of five classes (*15-1*). This setting is similar to the previous one except that the last 5 classes are learned sequentially, one by one. From Table 1 we can observe that performing multiple steps is challenging and existing methods work poorly for this setting, reaching performance inferior to 7% on both old and new classes. In particular, FT and prior-focused methods are unable to prevent forgetting, biasing their prediction completely towards new classes and demonstrating performances close to 0% on the first 15 classes. Even data-focused methods suffer a dramatic loss in performances in this setting, decreasing their score from the single to the multi-step scenarios of more than 50% on all classes. On the other side, our method is still able to achieve good performances. Compared to the other approaches, MiB outperforms all baselines by a large margin in both old (46.2% on the *disjoint* and 35.1% on the *overlapped*), and new (nearly 13% on both setups) classes. As the overall performance drop (11% on all classes) shows, the *overlapped* scenario is the most challenging one since it does not impose any constraint on which classes are present in the background.

Ablation Study. In Table 3 we report a detailed analysis of our contributions, considering the *overlapped* setup. We start from the baseline LwF [18] which employs standard cross-entropy and distillation losses. We first add to the baseline our modified cross-entropy (*CE*): this increases the ability to preserve old knowledge in all settings without harming (*15-1*) or even improving (*19-1*, *15-5*) performances on the new classes. Second, we add our distillation loss (*KD*) to the model. Our *KD* provides a boost on the performances for both old and new classes. The improvement on old classes is remarkable, especially in the 15-1 scenario (*i.e.* 22.8%). For the novel classes, the improvement is constant and is especially pronounced in the 15-5 scenario (7%). Notice that this aspect is peculiar of our *KD* since standard formulation work only on preserving old knowledge. This shows that the two losses provide mutual benefits. Finally, we add our classifiers' initialization strategy (*init*). This component provides an improvement in every setting, especially on novel classes: it doubles the performance on the *19-1* setting (22.1% vs 11.9%) and triplicates on the *15-1* (4.5% vs 13.5%). This confirms the importance of accounting for the background shift at the initialization stage to facilitate the learning of new classes.

4.4. ADE20K

ADE20K [39] is a large-scale dataset that contains 150 classes. Differently from Pascal-VOC 2012, this dataset contains both stuff (*e.g.* sky, building, wall) and object classes. We create the incremental datasets $\mathcal{T}^t$ by splitting the whole dataset into disjoint image sets, without any constraint except ensuring a minimum number of images (*i.e.* 50) where classes on $\mathcal{C}^t$ have labeled pixels. Obviously, each $\mathcal{T}^t$ provides annotations only for classes in $\mathcal{C}^t$ while other classes (old or future) appear as background in the ground truth. In Table 2 we report the mean IoU obtained averaging the results on two different class orders: the order proposed by [39] and a random one. In this experiments, we compare our approach with data-focused methods only (*i.e.* LwF, LwF-MC, and ILT) due to their gap in performance with prior-focused ones.

Single-step addition of 50 classes (*100-50*). In the first experiment, we initially train the network on 100 classes and we add the remaining 50 all at once. From Table 2 we can observe that FT is clearly a bad strategy on large scale settings since it completely forgets old knowledge. Using a distillation strategy enables the network to reduce the catastrophic forgetting: LwF obtains 21.1% on past classes, ILT

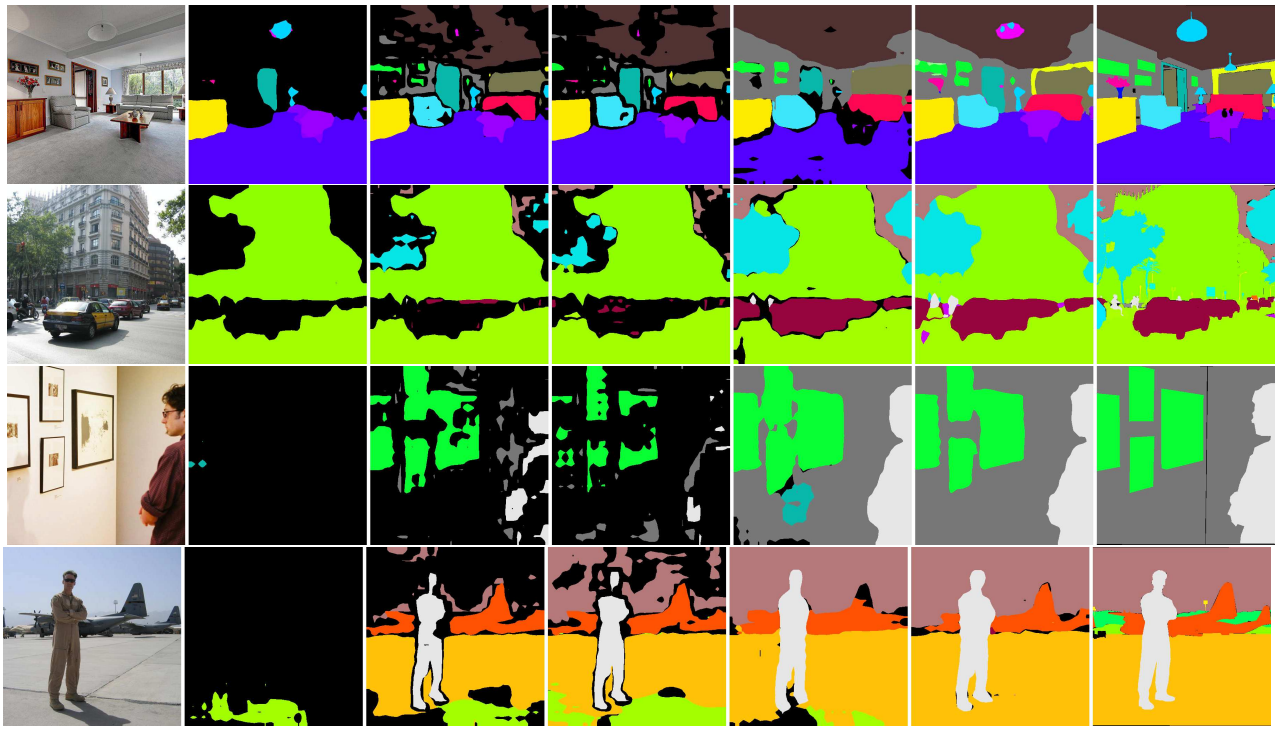

Figure 3: Qualitative results on the *100-50* setting of the ADE20K dataset using different incremental methods. The image demonstrates the superiority of our approach on both new (*e.g. building*, *floor*, *table*) and old (*e.g. car*, *wall*, *person*) classes. From left to right: image, FT, LwF [18], ILT [24], LwF-MC [28], our method, and the ground-truth. Best viewed in color.

22.9%, and LwF-MC 34.2%. Regarding new classes, LwF is the best strategy, exceeding LwF-MC by 18.9% and ILT by 6.6%. However, our method is far superior to all others, improving on the first classes and on the new ones. Moreover, we can observe that we are close to the joint training upper bound, especially considering new classes, where the gap with respect to it is only 0.3%. In Figure 3 we report some qualitative results which demonstrate the superiority of our method compared to the baselines.

Multi-step addition of 50 classes (*100-10*). We then evaluate the performance on multiple incremental steps: we start from 100 classes and we add the remaining classes 10 by 10, resulting in 5 incremental steps. In Table 2 we report the results on all sets of classes after the last learning step. In this setting the performance of FT, LwF and ILT are very poor because they strongly suffers catastrophic forgetting. LwF-MC demonstrates a better ability to preserve knowledge on old classes, at the cost of a performance drop on new classes. Again, our method achieves the best trade-off between learning new classes and preserving past knowledge, outperforming LwF-MC by 11.6% considering all classes.

Three steps of 50 classes (*50-50*). Finally, in Table 2 we analyze the performance on three sequential steps of 50 classes. Previous ICL methods achieve different trade-offs between learning new classes and not forgetting old ones. LwF and ILT obtain a good score on new classes, but they forget old knowledge. On the contrary, LwF-MC preserves knowledge on the first 50 classes without being able to learn new ones. Our method outperforms all the baselines by a large margin with a gap of 11.9% on the best performing baseline, achieving the highest mIoU on every step. Remarkably, the highest gap is on the intermediate step, where there are classes that we must both learn incrementally and preserve from forgetting on the subsequent learning step.

5. Conclusions

We studied the incremental class learning problem for semantic segmentation, analyzing the realistic scenario where the new training set does not provide annotations for old classes, leading to the semantic shift of the background class and exacerbating the catastrophic forgetting problem. We address this issue by proposing a novel objective function and a classifiers' initialization strategy which allows our network to explicitly model the semantic shift of the background, effectively learning new classes without deteriorating its ability to recognize old ones. Results show that our approach outperforms regularization-based ICL methods by a large margin, considering both small and large scale datasets. We hope that our problem formulation, our approach and our extensive comparison with previous methods will encourage future works on this novel research topic.

References

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

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE T-PAMI*, 39(12):2481–2495, 2017. 2

[3] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, 2018. 1, 2, 3, 4

[4] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, 2018. 1, 2, 5, 6

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE T-PAMI*, 40(4):834–848, 2017. 1, 2

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. 2017. 1, 2, 6

[7] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016. 1

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 1, 2

[9] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. Continual learning: A comparative study on how to defy forgetting in classification tasks. 2019. 2, 6

[10] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *CVPR*, 2019. 2

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 1, 2, 6

[12] Golnaz Ghiasi and Charless C Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *ECCV*, 2016. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. 2015. 1, 2, 4

[15] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, 2019. 1, 2

[16] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler L Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *AAAI*, 2018. 1

[17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 1, 2, 3, 5, 6

[18] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE T-PAMI*, 40(12):2935–2947, 2017. 1, 2, 3, 4, 5, 6, 7, 8

[19] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017. 1, 2

[20] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 2

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

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

[23] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 1, 2

[24] Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. In *ICCV-WS*, pages 0–0, 2019. 2, 4, 5, 6, 7, 8

[25] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *CVPR*, 2019. 2

[26] Firat Ozdemir, Philipp Fuernstahl, and Orcun Goksel. Learn the new, keep the old: Extending pretrained models with new anatomy and images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 361–369, 2018. 2

[27] Firat Ozdemir and Orcun Goksel. Extending pretrained segmentation networks with additional anatomical structures. *International journal of computer assisted radiology and surgery*, pages 1–9, 2019. 2

[28] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017. 1, 2, 3, 4, 5, 6, 7, 8

[29] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *CVPR*, 2018. 6

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

[31] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NeurIPS*, 2017. 2, 3

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

[33] Onur Tasar, Yuliya Tarabalka, and Pierre Alliez. Incremental learning for semantic segmentation of large-scale remote sensing data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(9):3524–3537, 2019. 2

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

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

[36] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017. 2, 5, 6

[37] Zhenli Zhang, Xiangyu Zhang, Chao Peng, Xiangyang Xue, and Jian Sun. Exfuse: Enhancing feature fusion for semantic segmentation. In *ECCV*, 2018. 1, 2

[38] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1, 2

[39] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 1, 2, 7