Learning New Classes from Limited Data in Image Segmentation and Object Detection

Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Computer and Control Engineering (35^{th}cycle)

Learning New Classes from Limited Data in Image Segmentation and Object Detection

By

Fabio Cermelli

******

Supervisor:
Prof. Barbara Caputo

Doctoral Examination Committee:
Prof. Bernt Schiele, Referee
Prof. Diane Larlus, Referee

Politecnico di Torino
2023

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Fabio Cermelli
2023

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

Abstract

Image segmentation is a critical capability for autonomous systems to understand their surroundings. Although deep neural networks have enhanced image segmentation performance, they require expensive and massive datasets for training, and they cannot update their knowledge for new classes without experiencing catastrophic forgetting. In the first part of this thesis, we address the issue of catastrophic forgetting by analyzing a unique aspect of semantic segmentation that exacerbates it. At each training step, the annotation only covers the new classes, while other classes, such as the ones already learned by the model, may appear in the image as background. To solve this, we propose a simple yet efficient solution that revisits the knowledge distillation framework and explicitly models this peculiarity. We also extend this approach to incremental learning in object detection and instance segmentation. In the second part of the thesis, we investigate learning new classes by reducing the number of images needed. We introduce the incremental few-shot semantic segmentation setting, where the model must learn new classes using only a few images. We propose a method for this novel setting that combines prototype learning and knowledge distillation, effectively preventing the model from forgetting old classes and overfitting the few images. Additionally, we explore the extreme setting where no labels are available for novel classes, proposing a self-training solution that extracts supervision from the unlabeled pixels in the training set. Finally, in the last part of the thesis, we aim to learn a segmentation model without relying on expensive pixel-level annotations, using cheaper alternatives instead. We suggest a general loss function to learn from points and scribbles, exploiting the assumption that all pixels in the image must belong to one of the annotated classes. Furthermore, we investigate the use of image-level labels for incremental learning in semantic segmentation. We present a new setting where a pre-trained model is trained to predict new classes using image-level labels. Building on the knowledge distillation framework, we propose an approach that integrates a localizer to extract pixel-level pseudo-supervision from image-level labels, which trains the model on novel classes without forgetting old ones.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

A long-standing dream of researchers and engineers is to develop agents that are capable of autonomously interacting with and operating in the world. One of the key abilities that autonomous agents must possess is the ability to perceive their surroundings and collect information from their sensors. Visual cameras, like human eyes, are crucial sensors that enable systems to collect visual data to extract information about the objects around them, their properties, and their functionalities.

Furthermore, autonomous systems should not only have a deep understanding of each object appearing in the image, but also of the scene as a whole, as it allows them to reason about the different objects and their relations. This ability is crucial in a wide variety of applications such as self-driving cars, medical image analysis, and surveillance systems.

Due to its practical importance in the real world, a lot of effort has been devoted to the field of computer vision with the goal of effectively analyzing the content and extracting information from images and videos. Researchers have focused on the image semantic segmentation task to enable systems to perceive their environment and reason about the scene effectively. This task involves dividing an image into different regions and assigning each pixel to a specific category or class, providing information on both the objects themselves (their location, shape, and size) and their relations (their relative position, interaction, etc.).

In recent years, there has been a steep improvement in the semantic segmentation task, thanks to the availability of large collections of images from the web [43, 84, 202, 32, 109], and the development of advanced deep neural network architectures [96, 27, 149, 186, 31]. In particular, Fully Convolutional Neural Networks [96, 27] are the leading paradigm for semantic segmentation, being highly effective in extracting visual features from images at high resolution and providing precise information for each pixel in the image. However, they have recently been challenged by visual transformer architectures [149, 186, 31], which provide the additional feature of relating distant pixels, even from the first network layers.

Despite their effectiveness in image segmentation, deep neural networks still have some limitations. First, they are extremely data-hungry and require thousands of annotated images for training. This is a general issue for any computer vision task, but it is even more severe in semantic segmentation, where each image should be annotated at the pixel level, i.e., a class label should be provided for each pixel of the image. This leads to a prohibitive cost for obtaining the dataset and limits the applications in the real world. Another major drawback of deep architectures is that they are not designed to update their knowledge when new categories are discovered and are limited to the set of classes present in the initial training

set. Despite best efforts to collect the dataset, it is often hard to predetermine all the classes that will be seen during system operation since the world is constantly changing, and new devices, objects, and even diseases may appear every day.

Incremental Learning.

One possible solution to the latter problem could be to supplement existing datasets with additional samples and retrain models from scratch. However, this approach may prove impractical for scenarios involving frequent updates, as training on the expanded dataset could be too time-consuming, leading to increased energy usage and carbon emissions by machine learning models [119, 148, 160]. Moreover, retraining may not be viable if the original data is no longer accessible due to privacy or intellectual property concerns.

An alternative approach that could be more effective is to fine-tune the model solely on the extra annotated samples related to the new categories, enabling quicker and less expensive updates. However, deep neural networks face a challenge in updating their parameters for new classes without catastrophic forgetting [102], which erases previous knowledge.

The main goal of incremental learning is to develop a solution that can mitigate the problem of catastrophic forgetting encountered in the latter approach. While research efforts have traditionally focused on image classification [81, 132, 139, 69, 39, 41] and object detection [145, 120], less attention has been paid to the semantic segmentation task.

In the first part of the thesis, we propose to fill this research gap by investigating the additional challenges posed by the task. Specifically, we examine the scenario where a model needs to be updated to segment new classes, given a dataset containing pixel-level annotations only for them. However, the images may also contain classes outside the new ones that are not annotated, either being classes seen in previous updates or that will be seen in the future. The pixels of these classes are then considered as belonging to the special background class, which contains all the pixels for which annotation is not provided. This unique aspect introduces a challenge: the semantics of the background class change with every training step, exacerbating catastrophic forgetting. Without adequately modeling the semantic shift of the background class, the model may classify all old classes as belonging to the background class, causing it to forget previous knowledge after a few iterations.

We propose a simple yet effective solution to this problem by revisiting the standard knowledge distillation framework [81, 132] to consider that the background may contain either old or future classes. Additionally, we extend our solution to incremental learning in object detection and instance segmentation, where the issue of missing annotations for old

and future classes has been overlooked by previous works [145, 120], but has a similar effect on model performance, leading to severe catastrophic forgetting.

Although incremental learning effectively enables the integration of new classes over time, it still requires the collection and annotation of a large dataset for the novel classes, where labels are provided for each pixel, resulting in prohibitive expenses for practical applications. To effectively alleviate the burden of collecting the dataset and learning novel classes in a data-efficient manner, two different directions can be investigated: (i) designing models that can learn novel classes using only a limited number of pixel-level annotated images, and (ii) avoiding the use of expensive pixel-level annotations by resorting to weaker annotation types, such as points, scribbles, or image-level labels.

Few-Shot and Zero-Label Semantic Segmentation.

The former direction aims to emulate the human ability to quickly learn novel classes by associating them with the ones already known. However, such an ability is still an open challenge for semantic segmentation models as the process requires a strong knowledge transfer between old and new classes while preventing catastrophic forgetting. In the research literature, this problem has been addressed in two separate fields: few-shot [143, 128, 146, 185] and zero-label semantic segmentation [185, 15, 51].

In few-shot semantic segmentation, the task is to learn novel classes with only a few labeled examples. Previous works have either considered the setting as a binary segmentation problem [143, 128, 37, 174], i.e., focused only on segmenting a single novel class, or allowed the use of all existing images to fine-tune the model on the novel classes [146, 185], which is often an unrealistic assumption. In the second part of the thesis, we introduce a novel and more realistic setting, named incremental few-shot semantic segmentation, with the goal of extending a segmentation model to learn new classes with only a few annotated images and without relying on previously seen data. We propose a framework that addresses the challenges of the novel task by combining prototype learning [125, 47] and knowledge distillation [81, 132] to learn novel classes without overfitting the few images and forgetting the old knowledge.

The zero-label semantic segmentation setting involves learning new classes without any annotation for them. A common solution employed by previous works [185, 15, 51] is to exploit textual descriptors, such as word embeddings, for all the classes and force the network to learn a mapping between the visual features and the word space. However, previous works completely ignore the fact that novel classes may also appear in the training dataset but without annotations. In this thesis, we propose to exploit these unlabeled pixels and introduce

a novel method that employs a self-training pipeline to provide labels for them. To reduce the noise in the pseudo-labels, we enhance the supervision with a consistency constraint that filters out predictions not consistent across different image augmentations.

Weakly Supervised Learning.

The second solution to reduce the annotation burden is to avoid pixel-level labels and consider alternative forms of supervision. To this aim, different types of annotations have been exploited to learn a semantic segmentation model, such as bounding boxes [33, 68], scribbles [82, 167], points [11], and image-level labels [72, 118, 124, 8, 2, 77].

Points and scribbles are effective labels since they are cheap and provide precise localization information on the target classes. They only require the annotator to draw a point or a line on each class in the image. However, previous works [11, 126, 82] only considered the few annotated pixels, disregarding the information that can be obtained from all the others. In this thesis, we exploit the assumption that all the pixels in the image must belong to one of the classes in the annotation. Specifically, we derive a general loss function that can extract supervision from them by maximizing the probability of having any of the classes appearing in the image in every pixel. Despite being simple, this loss demonstrates to achieve results comparable and even superior to hand-crafted methods introduced by previous works.

Image-level labels are the cheapest annotation type, only requiring the annotator to report the classes appearing in the image without providing any localization cue. They are also easy to collect from the web, obtaining them from search engines or from the numerous available classification datasets, such as ImageNet [35]. Despite their affordability, training segmentation models with image-level labels is challenging since the model has to extract localization cues by itself. Previous works [72, 2, 77] proposed to employ a classification model trained using image-level labels to extract pixel-level pseudo-labels that are then used to train a downstream segmentation model. However, they all focused on offline scenarios where the model has a fixed knowledge of the world. In the third part of this thesis, we take a different direction and investigate how to extend a segmentation model to learn new classes over time using only cheap image-level labels. We propose a novel method that builds on the knowledge distillation framework [81, 132] and introduces a localizer to extract pseudo-supervision from the weak labels to learn novel classes.

1.2 Contributions

Overall, the goal of this thesis is to provide effective and innovative solutions to extend semantic segmentation models to new classes in a data-efficient manner without forgetting previous knowledge. Specifically, in the first part, we introduce the task of incremental learning in semantic segmentation, outlining the additional challenges of this scenario and proposing a simple yet efficient solution. In the second part, we investigate the use of a limited number of labeled images to learn new classes and propose the incremental few-shot segmentation setting. We also explore the extreme case where no annotation is provided for novel classes, presenting a novel method for zero-label semantic segmentation. Lastly, in the final part, we examine how to learn to segment novel classes without requiring pixel-level supervision by exploring weaker types of annotations such as points, scribbles, and image-level labels. Below is a detailed list of the contributions presented in this thesis.

Incremental Learning. We investigate incremental learning for semantic segmentation, object detection, and instance segmentation tasks, introducing the following contributions.

- We present the first benchmark for incremental learning in semantic segmentation that considers the peculiar distribution shift issue that arises due to the presence of the background class. The benchmark considers several previous incremental learning methods proposed for image classification on two popular semantic segmentation datasets.
- We introduce an incremental learning framework for semantic segmentation that is able to cope with the semantic shift of the background class. In particular, it proposes to revisit the classic objective function of incremental learning, the cross-entropy loss, and the knowledge distillation loss, to explicitly model the evolving semantics of the background class.
- We extend the previous method for object detection and instance segmentation, addressing a similar issue arising from the missing annotations of the old and future classes in the current learning step.

Few-Shot and Zero-Label Segmentation. In the context of learning to segment new classes from a limited number of images, we will introduce the following contributions.

- We introduce the few-shot incremental semantic segmentation setting. Differently from previous settings, we focus on learning new classes from few-annotated images, without forgetting old classes and with no access to the old training dataset.

- We present a framework to learn new classes from a few annotated samples. The framework is made in two steps: first, it computes the class prototypes for the new classes and injects them into the classification weights, then, it fine-tunes the whole network on the few images by using a tailored knowledge distillation framework.
- We investigate the limits of current zero-label semantic segmentation methods and we show that performing self-training introducing a consistency constraint largely improves the performance.

Weakly-Supervised Segmentation. We investigate three types of weak annotations, proposing the following contributions.

- We propose a novel technique to learn from point and scribble supervision by exploiting the unlabeled pixels. In particular, relying on the assumption that they should belong to the background or to a class for which annotation is present in the image, we propose a novel loss function that maximizes the probability on each unlabeled pixel of having either an annotated class or the background.
- We introduce the weakly-supervised incremental learning semantic segmentation setting (WILS), where we assess the abilities of methods to learn to segment new classes over time, without forgetting, being provided only a dataset containing image-level labels for new classes.
- We are introducing a novel framework for WILS, which we have name WILSON. This framework includes a localizer module that extracts pseudo-labels for new classes using image-level labels. The localizer module is also regularized to obtain more accurate object boundaries by utilizing a localization-prior from the segmentation network. Once the pseudo-labels are obtained, we employ a knowledge-distillation technique to train the current network on the new classes without forgetting, by combining supervision from both the localizer and the old network. This approach enhances the accuracy of the current network in recognizing the new classes.

1.3 Outline

In Chapter 2, we will introduce the main concepts of semantic segmentation and incremental learning. We will provide an definition of semantic segmentation, introducing a formal problem statement and the metric used for the task. In addition, we will provide an overview of the most relevant works in semantic segmentation, with a particular focus on the Deeplab architecture [26–28], and of the most common datasets for semantic segmentation. Next, we will formally introduce the task of incremental learning, providing a summary of its main setting definions. Finally, we will provide an overview of the incremental learning literature.

Chapter 3 will discuss the challenges of incremental learning in semantic segmentation and object detection. In Section 3.2, we will describe the first method for incremental learning in semantic segmentation (ILSS). In particular, we initially formalize the problem of ILSS, highlighting the semantic shift of the background class in consecutive incremental training steps. Next, we describe the **MiB** (Modeling the Background) framework, which revisits the classical knowledge distillation framework modeling the semantic shift of the background. In Section 3.3, we investigate the problem of incremental learning in the object detection and instance segmentation task, noting that, similarly to Sec. 3.2, since the annotations for old and future classes in a learning step are missing, the catastrophic forgetting is exacerbated. To solve the issue, we propose the **MMA** (Modeling the Missing Annotation) method, which revisits the standard knowledge distillation losses to keep the missing annotation into account.

Chapter 4 introduces works enabling learning to segment new classes with few or zero images. First, we introduce a benchmark for few-shot incremental learning in semantic segmentation in Sec. 4.2 and a framework able to work in such a challenging setting. In particular, we present **PIFS** (Prototype-based Incremental Few-Shot Segmentation) that combines prototype learning with the knowledge distillation paradigm, preventing overfitting on the new classes while avoiding forgetting. Next, Section 4.3 analyzes the extreme case where no images are provided for novel classes and the model should learn to segment them by being provided only a textual descriptor. Investigating the current state-of-the-art, we note that they are not considering that unlabeled pixels contain complementary information about unseen classes and thus we present **STRICT**, a method based on self-training to exploit them. Self-training is performed by generating pseudo-labels that are further refined to respect a consistency constraint among different augmented versions of the same image.

Chapter 5 focuses on learning a segmentation model using weak annotations. In Sec. 5.2, we present a simple method that is able to learn a segmentation model from point and scribble annotations. While on the annotated pixels we can use a partial cross-entropy loss,

we demonstrate that it is beneficial to use the other pixels for training. We design a loss exploiting the assumption that, while they are not annotated, they can only contain one of the classes in the annotation. Furthermore, in Sec. 5.3, a method learning to segment new classes over time using cheap and widely available image-level is presented. The section first introduces the novel setting, WILS (weakly incremental learning in semantic segmentation), and then presents a framework, named **WILSON** to address it. To learn to segment new classes, WILSON introduces a localizer network that provides pseudo-labels for the new classes. Moreover, to avoid forgetting, it exploits the predictions of the network trained in previous incremental steps. Mixing the pseudo-labels and the prediction of the old model, we design a novel learning objective that learns new classes, without forgetting.

Finally, in Chapter 6, we summarize the findings of the thesis and identify the open problems and future research directions.

1.4 Publications

In the following section, the author's publications are listed. Note that some articles have not been included in the thesis. The included article are reported in bold.

- Mancini, M., Porzi, L., Cermelli, F., and Caputo, B. (2019).
  Discovering latent domains for unsupervised domain adaptation through consistency.
  In International Conference on Image Analysis and Processing (pp. 390-401). Springer, Cham.
- Cermelli, F., Mancini, M., Ricci, E., and Caputo, B. (2019).
  The RGB-D triathlon: Towards agile visual toolboxes for robots.
  In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 6097-6104). IEEE.
- Cermelli, F., Mancini, M., Bulo, S. R., Ricci, E., and Caputo, B. (2020).
  Modeling the background for incremental learning in semantic segmentation.
  In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 9233-9242).
- Fontanel, D., Cermelli, F., Mancini, M., Bulo, S. R., Ricci, E., and Caputo, B. (2020).
  Boosting deep open world recognition by clustering.
  IEEE Robotics and Automation Letters, 5(4), 5985-5992.
- Fontanel, D., Cermelli, F., Mancini, M., and Caputo, B. (2020).
  On the challenges of open world recognition under shifting visual domains.
  IEEE Robotics and Automation Letters, 6(2), 604-611.
- Pastore, G., Cermelli, F., Xian, Y., Mancini, M., Akata, Z., and Caputo, B. (2021).
  A closer look at self-training for zero-label semantic segmentation.
  In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (Workshop) (pp. 2693-2702).
- Fontanel, D., Cermelli, F., Mancini, M., and Caputo, B. (2021).
  Detecting anomalies in semantic segmentation with prototypes.
  In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (Workshop) (pp. 113-121).
- Cermelli, F., Mancini, M., Xian, Y., Akata, Z., and Caputo, B. (2021).
  Prototype-based incremental few-shot segmentation.
  In British Machine Vision Conference (BMVC 2021).

- Cermelli, F., Mancini, M., Bulo, S. R., Ricci, E., and Caputo, B. (2021).
  Modeling the background for incremental and weakly-supervised semantic segmentation.
  IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.44, no.12, pp. 10099-10113 (2021).
- Tavera, A., Cermelli, F., Masone, C., and Caputo, B. (2022).
  Pixel-by-pixel cross-domain alignment for few-shot semantic segmentation.
  In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 1626-1635).
- Arnaudo, E., Cermelli, F., Tavera, A., Rossi, C., and Caputo, B. (2022).
  A contrastive distillation approach for incremental semantic segmentation in aerial images.
  In International Conference on Image Analysis and Processing (pp. 742-754). Springer, Cham.
- Cermelli, F., Fontanel, D., Tavera, A., Ciccone, M., and Caputo, B. (2022).
  Incremental learning in semantic segmentation from image labels.
  In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4371-4381).
- Cermelli, F., Geraci, A., Fontanel, D., and Caputo, B. (2022).
  Modeling Missing Annotations for Incremental Learning in Object Detection.
  In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (Workshop) (pp. 3700-3710).
- Fontanel, D., Cermelli, F., Geraci, A., Musarra, M., Tarantino, M., and Caputo, B. (2022).
  Relaxing the Forget Constraints in Open World Recognition.
  In International Conference on Image Analysis and Processing (pp. 751-763). Springer, Cham.
- Fantauzzo, L., Fani, E., Caldarola, D., Tavera, A., Cermelli, F., Ciccone, M., and Caputo, B. (2022).
  FedDrive: Generalizing Federated Learning to Semantic Segmentation in Autonomous Driving.
  In 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE.

Chapter 2

Preliminaries

2.1 Semantic Segmentation

Semantic segmentation aims to partition an image into regions corresponding to the same category, allowing for the extraction of meaningful information from the image and the identification of different regions within the image. Semantic segmentation can be considered a pixel-level classification problem, where the goal is to assign a class label to each pixel in the image, such as “car" or “sky", regardless of distinguishing different instances of the same class (*e.g.* two different cars belong to the same "car" segment).

Problem Statement. Formally, let us denote as $\mathscr{X}$ the input space (*i.e.* the image space) and, without loss of generality, let us assume that each image $x \in \mathscr{X}$ is composed by a set of pixels $\mathscr{I}$ with constant cardinality $|\mathscr{I}| = N$. The output space is defined as $(\mathscr{Y})^N$, with the latter denoting the product set of N-tuples with elements in a label space $\mathscr{Y}$. Given an image x the goal of semantic segmentation is to assign each pixel x_i of image x a label $y_i \in \mathscr{Y}$, representing its semantic class. Out-of-class pixels can be assigned a special class, *i.e.* the background class $\mathsf{b} \in \mathscr{Y}$. Given a training set $\mathscr{T} \subset \mathscr{X} \times (\mathscr{Y})^N$, the mapping is realized by learning a model f_θ with parameters θ from the image pixels space $\mathscr{X}$ to their corresponding segmentation mask $(\mathscr{Y})^N$, *i.e.* $f_\theta : \mathscr{X} \mapsto (\mathscr{Y})^N$.

Metrics. The easiest way to evaluate the model performance is to extend the standard metric of image classification to pixel-level, *i.e.* using the Pixel Accuracy (PA). This metric measures the number of pixels correctly classified. However, due to the large unbalance in the class distribution, PA is largely biased toward the most frequent or larger classes and fails to consider smaller objects. To better represent object of different sizes and with different frequency, the mean Intersection over Union (mIoU) [43], or Jaccard Index, has been introduced. Specifically, the Intersection over Union (IoU) computes the overlap between the predicted segment for a class and the corresponding ground-truth annotation, divided by their union, as illustrated in Fig. 2.1. Formally, the IoU is defined as:

$$IoU(c) = \frac{TP(c)}{TP(c) + FP(c) + FN(c)}, \tag{2.1}$$

where TP, FP, and FN represent, respectively the true positive, false positive, and false negative for the class c. When the prediction perfectly matches the annotation, the intersection corresponds to the union, and the IoU score equals one. Differently, if the intersection is void, the IoU score equals zero. The mIoU is then obtained by computing the average of the IoU for all the classes considered.

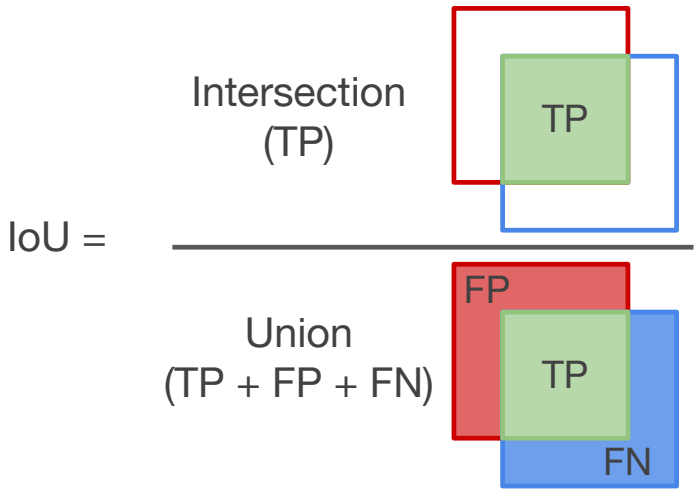


Fig. 2.1 Illustration of the Intersection over Union (IoU) metric.

2.1.1 Literature Review

Early works on semantic segmentation addressed the task as a clustering problem, introducing additional information from edges and contours [62, 179]. With the rise of deep learning, the region-based segmentation approaches [16, 49] have been introduced. First, they extract distinct, free-form regions from an image and describe them using various features. Then, they use these descriptions to classify the regions into different classes. At test time, the region-based predictions are transformed into pixel-level predictions by assigning each pixel to the class of the region it belongs to. This process is known as *segmentation using recognition*. Fully-convolutional networks (FCN) [96], however, replaced them by considering semantic segmentation as a per-pixel classification problem, learning a mapping from pixels to classes without the use of region proposals. To improve the output sharpness, some works [137, 96] proposed to mix the features coming from the first layers of the networks with the high-level information coming from the last layers. Other works introduced network modules to aggregate long-range dependencies in the feature maps [26–28, 200] or to exploit contextual information [45, 61, 193, 192]. In the last few years, transformer [38] architectures are getting large interest from the community due to their ability to incorporate long-range dependencies at every layer of the network and many promising architectures have been proposed [149, 186, 66, 94].

DeepLab [26–28]. In this thesis, we will frequently use DeepLab as a backbone for our studies due to its simplicity and effectiveness. DeepLab [26–28] is a series of state-of-the-art fully-convolutional network models developed for semantic segmentation. The architecture of the third version, illustrated in Fig. 2.2, is based on two parts: a feature extractor and a

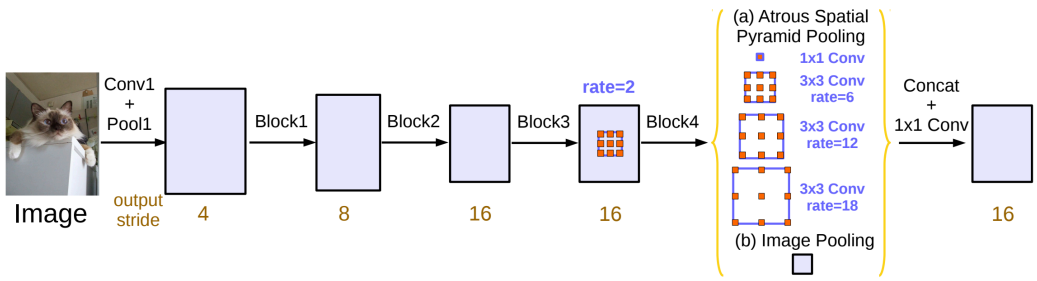


Fig. 2.2 Overview of Deeplab-v3 [27] architecture. Image is taken from [27].

segmentation head. The feature extractor can be any classification backbone, often a ResNet [55], whose output resolution is increased thanks to the use of dilated (atrous) convolutions [190] that allow the model to aggregate more long-range dependencies without increasing the number of model parameters. DeepLab-v3 designs a special segmentation head that is called Atrous Spacial Pyramidal Pooling (ASPP). It is constituted by multiple parallel convolutions with different dilation rates to capture multi-scale context and a global pooling operation, which are then concatenated and processed to obtain the final classification prediction.

2.1.2 Datasets

Semantic Segmentation has seen large improvements in the last years thanks to the publication of novel datasets that have been collected in multiple contexts and for different target applications, such as automous driving [32, 6, 109, 136] and earth observation [172, 13]. In this thesis, we will employ three widely used datasets that contains images taken from common contexts.

Pascal-VOC 2012 (VOC) [43] is a benchmark that contains 10582 training and 500 validation images taken from common scenes. It includes 20 object categories, including vehicles, animals, and indoor objects, and an additional *background* class consting of all the non-annotated pixels. Following the common practice, in this thesis we will include the pixel-level annotations from the SBD dataset [54]. Fig. 2.3a reports some examples of VOC images.

ADE20K [202] is a challenging benchmark that contains 20 thousands images taken from complex urban and indoor scenes. The datasets contains 150 classes, including both thing (vehicles, indoor and street objects) and stuff (such as *sky*, *grass*, *wall*) classes, with very different visual appearance, shape, and frequency. Some examples of the dataset are reported in Fig. 2.3b.

(a) Pascal-VOC 2012 (b) ADE20K (c) COCO-Stuff

Fig. 2.3 Illustration of semantic segmentation datasets.

COCO [18] and **COCO-Stuff** [17] are large-scale benchmarks for multiple tasks, including semantic segmentation. They contain the same 164 thousands images but differs in the annotation: COCO only contains 80 object (*thing*) classes while COCO-Stuff extends them with 91 additional *stuff* classes. The *stuff* pixels in COCO are unified in a single *background* class. Fig. 2.3c reports examples of COCO-Stuff.

2.2 Incremental Learning

Incremental learning (also called continual or life-long learning) studies the ability of neural network of incorporating new knowledge over time. This is a challenging problem in deep learning as traditional neural networks are not designed to retain previous knowledge while learning on a changing distribution, suffering catastrophic forgetting [102]. A simple solution to avoid forgetting would be to maintain a constantly growing datasets containing all the knowledge learned so far and retrain the model from scratch every time new data is discovered. However, this is highly impractical for two reasons: first, it would require an ever increasing memory to store all the data, and second, it would require more and more computational resources to re-train the model from scratch every time. In addition, there may be use-case where the training data are protected by privacy or intellectual property constraint and cannot be used when retraining the model. Incremental learning thus focus on extending the knowledge by fine-tuning the model on new data, representing the knowledge to be incorporated, and possibly a small sample of previous data.

Three incremental learning settings can be found in literature [165], that are distinguished on the type of knowledge to be integrated at every step: domains, tasks, or classes.

- Domain-incremental learning [100, 131, 169, 170] assesses the ability of a model to deal with the covariate shift, *i.e.* when the data distribution changes and new domains are discovered.
- Task-incremental learning [99, 81, 131] is closely related to multi-task learning, where a model must perform different tasks simultaneously. Task-incremental learning is approached as a multi-head setting, *i.e.* the tasks are considered independently and the model often has a separate head to predict the output of each task. At inference time, the models knows which is the task it has to perform.
- Class-incremental learning [132, 145, 21] adds, at every time step, new classes to the model. Differently from the previous scenarios, it is considered a single-head setting, *i.e.* the model should perform predictions among all the classes seen so far and it has a single classification head.

In this thesis, we will focus on the more realistic and practical class-incremental learning scenario, illustrated in Fig. 2.4.

Problem Statement. In class-incremental learning the training is performed over multiple phases, called *training steps*. Each training step t introduces a novel set of classes C^t

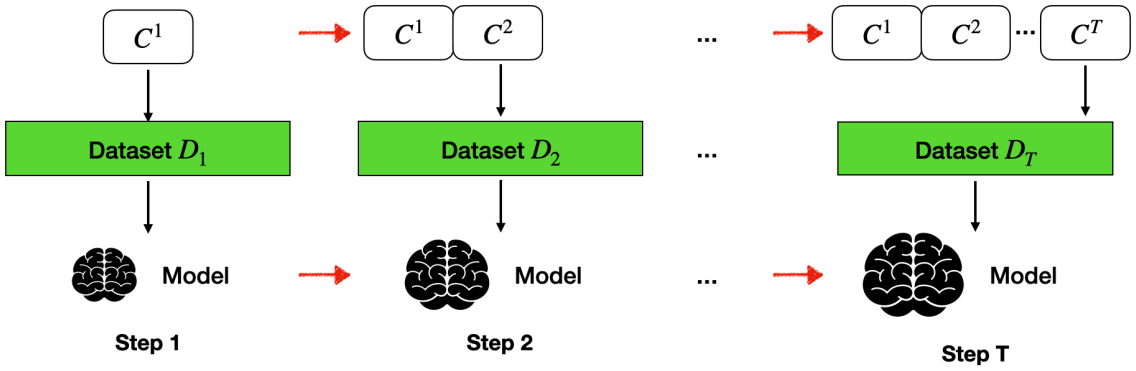


Fig. 2.4 An illustration of a class-incremental learning setting. At each training step t it is introduced a new set of classes C^t and provided a new dataset D_t containing labels for them. The model is required to learn the new classes while avoiding forgetting the old ones.

and a novel dataset $\mathscr{T}_t$, that contains images and labels for the novel classes. The goal of incremental learning is to update the model's parameters θ^{t-1}, trained at the step $t-1$, to obtain a set of parameters θ^t able to classify all the classes seen so far $Y^t = \bigcup_{s=1}^{t} C^s$. Note that the parameters θ^t are obtained on the new dataset $\mathscr{T}_t$, resulting in optimal parameters that may largely different from previous tasks, inducing catastrophic forgetting [102] of the previous classes. The challenge of incremental learning is find a good constrain for the optimization problem such that the new parameters are close to the optimum for the previous classes while able to learn the new classes.

2.2.1 Literature Review

Incremental-class learning has been extensively studied for the image classification task [69, 139, 132, 24, 180, 59, 39, 41]. We can group the incremental learning approaches into three categories: architecture-based [139, 187, 142], structural-based [69, 24, 194], and functional-based [81, 132, 20, 36].

Architecture-based methods [139, 187, 98, 99, 92] aim to mitigate catastrophic forgetting by modifying the neural network architecture to allocate more parameters for new tasks while retaining knowledge of previous tasks. Progressive Networks [139] introduces lateral connections to the main network at each incremental learning step, allowing the network to obtain useful features for new tasks while freezing the old parameters, thus avoiding catastrophic forgetting. However, this approach can result in a large network after multiple tasks. PackNet [98] is based on pruning techniques and performs an iterative pruning and fine-tuning procedure. At each training step, PackNet fine-tunes the free parameters of the

network on the new task and then removes redundant parameters through pruning, ensuring minimal performance drop. Piggyback [99] starts from a pre-trained network and learns a set of binary masks for each new task, without tuning the original parameters and introducing minimal memory overhead. However, both PackNet and Piggyback are typically evaluated on multi-task settings where the task-id is provided during inference [98, 99]. Recently, Dynamically Expandable Networks with Regularization (DER) [187] was proposed, which achieves state-of-the-art performance on single-head evaluations while maintaining a low memory footprint. DER adds a new feature extractor for each new training step and prunes the network aggressively after training to limit parameter explosion.

Structural-based methods aim to avoid catastrophic forgetting by constraining the new model to minimize differences in parameter values with respect to old ones [69, 194, 24, 7, 93]. Simply freezing the old model would not allow for learning novel classes, so these methods aim to identify and penalize changes in the parameters most important for old tasks while allowing other parameters to change to fit new tasks. The methods mostly differ in how parameter importance is computed. Elastic Weight Consolidation (EWC) [69] proposed to employ the diagonal Fisher information matrix to compute the importance of each parameter. In particular, high values in the matrix are important to learn the previous task and thus, their value should not change in the following training steps. Zenke et al. [194] proposed to compute the score by computing the path integral of the gradient vector field along the parameter trajectory, *i.e.* it estimates how each parameter contributed to changes in the total loss. Chaudhry et al. [24] proposed a generalization of the two previous works [69, 194]. MAS [7] estimates the importance of each parameter by measuring the sensitivity when perturbation are applied to them. While structural-based methods offer promising solutions to the issue of catastrophic forgetting in incremental learning scenarios, their scalability to large models and datasets is still challenging.

Functional-based methods [81, 39, 180, 132, 36] aim to prevent changes in the output space of the model, rather than in the parameter space. The majority of the methods in this group exploit knowledge distillation [57]. Specifically, at each training step, two instances of the model are maintained: one is frozen after the previous learning step and acts as a teacher, while the other is the student network that is trained on the new data. To prevent forgetting, a regularization term is added that minimizes the distance between the activations produced by the old network and the new one, effectively constraining the student to mimic the teacher model outputs. The foundational work of this category is Learning without Forgetting (LwF) [81], which proposes employing the Kullback-Leibler divergence between the probabilities of the old and new models. Subsequent works have built upon LwF to improve its performance

[132, 180, 59, 20]. Learning without Memorizing [36] proposed a different approach, which is to minimize the differences in the attention maps produced by GradCAM [141]. PodNet [39] proposed employing distillation also in the intermediate network features. Approaches based on knowledge distillation demonstrated good flexibility and scalability across a wide variety of tasks and settings, and thus, in this thesis, we will largely build methods belonging to this category.

Rehearsal. In order to prevent forgetting in incremental learning scenarios, one common technique is to use rehearsal learning. Rehearsal learning [132, 20, 59, 180] involves storing relevant samples of previous classes in a small memory, which can then be used in subsequent training steps. To ensure that the amount of old samples does not become too large, two main strategies have been proposed. ICaRL [132] suggest keeping a fixed number of samples per class, and decreasing the number of samples per class with each subsequent training step. LUCIR [59], on the other hand, propose keeping a fixed number of samples per class, and increasing the memory size with each step. The stored exemplars can either be used during the training of new classes, mixed with samples from the new dataset [59, 132], or used in a balanced fine-tuning procedure after each training step [20, 180]. Some studies prefer to avoid storing old samples altogether due to data privacy or intellectual property concerns, and instead use generative adversarial networks (GANs) [50] to generate images of old classes [112, 144]. Finally, Liu et al. [91] propose a different approach, where exemplars are parameterized and optimized on every training step to obtain the best representation for both new and old classes.

Transformer architectures have recently demonstrated outstanding performance in computer vision [38, 164, 163, 94] attracting the attention of the incremental learning community [41, 177, 176] and showing interesting properties to learning new classes over time. In particular, given the transformer use of classification token, DyTox [41] proposed to extract an ad-hoc classification token per task. Differently, Learning to Prompt [177] stores a pool of prompts that are employed to condition the whole forward execution of the patch tokens.

Chapter 3

Incremental Learning in Segmentation and Object Detection

3.1 Introduction

Incremental Learning has been extensively considered in image classification, but only a few works extended it to more complex vision tasks. Shmelkov et al. [145] proposed to extend incremental learning in object detection, while Michieli et al. [103] introduced a relaxed incremental learning setting in semantic segmentation. However, they did not accounted an additional challenge when bringing incremental learning to dense prediction tasks: the presence of multiple classes in every image. Images in the current step, in fact, report annotations only for new classes but they may contain also old or future classes. Without considering and modeling their presence, catastrophic forgetting is exacerbated, resulting in models unable to correctly predict old classes after a few incremental learning steps.

In this chapter, we first analyze the problem in semantic segmentation, showing that all the non-annotated classes are collapsed in the artificial background class, introducing the background-shift issue. We propose a method that revisits the standard knowledge distillation framework to keep into account the shift and effectively address the issue. To assess our contribution, we propose a novel semantic segmentation benchmark and we show our method, named MiB, outperforms all the previous works, achieving a new state of the art.

Next, we analyze the problem in the object detection and instance segmentation tasks. Similarly, we find that all the non-annotated objects are collapsed into background regions, aggravating catastrophic forgetting. We design an approach that revisits the common knowledge distillation technique to take into account the missing annotations. We benchmark our approach, named MMA, in the existing object detection setting [145] and in a novel instance segmentation setting, showing substantial performance gains with previous works.

The work presented in this chapter led to the publication of two works:

- Cermelli, F., Mancini, M., Bulo, S. R., Ricci, E., and Caputo, B. *Modeling the background for incremental learning in semantic segmentation*. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 9233-9242).

- Cermelli, F., Geraci, A., Fontanel, D., and Caputo, B. *Modeling Missing Annotations for Incremental Learning in Object Detection*. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (Workshop) (pp. 3700-3710).

3.2 Incremental Learning in Semantic Segmentation

Incremental Learning in Semantic Segmentation has been overlooked by the community despite being interesting and practical for applications. Previous works either investigate the task for particular domains, such as medical [115, 114] or satellite [158] images, or study the task in a relaxed setting [103], where the annotations for both old and new classes are available at every training iteration. In this section we fill this research gap, proposing a new realistic incremental class learning (ICL) setting for semantic segmentation where only new class annotations are considered at every step.

A particular characteristic of semantic segmentation is the existence of the background class, which identifies pixels that are not assigned to any other category. This class has a minimal impact on the design of traditional offline semantic segmentation methods but it is critical in the incremental learning setting. In fact, given that only the labels for the novel classes are available at every incremental step, all the other pixels in the image are considered background, either if they are old classes (seen in previous learning steps) or classes that will appear in future steps. This shift in the semantics of the background class, illustrated in Fig. 3.1, exacerbates the issue of catastrophic forgetting [102] and requires to be properly considered by the method design to maintain good performance.

Inspired by previous ICL works in image classification [81, 132], we design a method based on the knowledge distillation approach. However, we revisit the classical distillation-based framework for incremental learning [81] by introducing two novel loss terms to properly account for the semantic distribution shift within the background class, thus proposing the first ICL approach tailored to semantic segmentation. We extensively evaluate our method on two datasets, Pascal-VOC [43] and ADE20K [202], showing that our approach, coupled with a novel classifier initialization strategy, outperforms traditional ICL methods.

The contributions of this section are: (1) a study of incremental class learning for semantic segmentation and the problem of distribution shift due to the background class, (2) the proposal of the new classification and knowledge distillation objective functions and classifier initialization strategy to explicitly cope with the evolving semantics of the background class, and (3) a benchmark comparing our approach over several previous ICL methods on two popular semantic segmentation datasets, considering different experimental settings. The code to replice our results and the benchmark can be found at https://www.github.com/fcdl94/MiB.

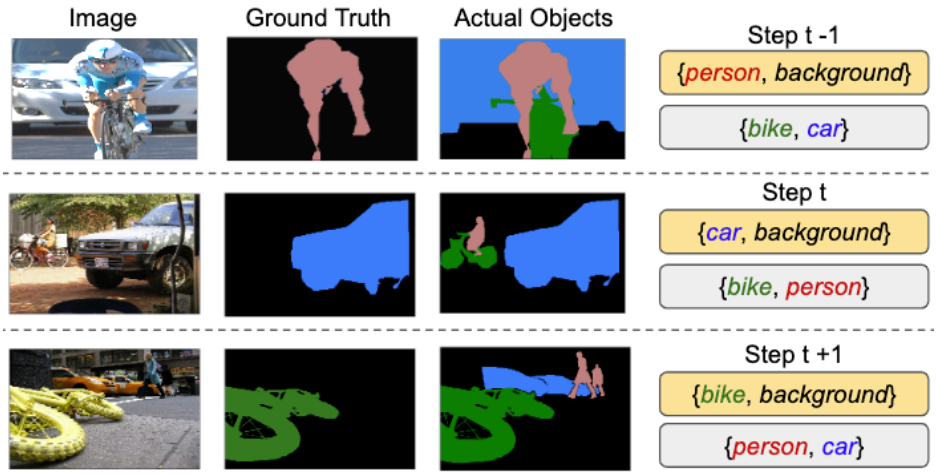


Fig. 3.1 Illustration of the semantic shift of the background class in incremental learning for semantic segmentation. Yellow boxes denote the ground truth provided in the learning step, while grey boxes denote classes not labeled. As different learning steps have different label spaces, at step t old 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.

3.2.1 Related Works

Most semantic segmentation techniques operate under an offline scenario [96, 28, 200, 83, 198, 27, 26], meaning that the training data for all classes is available beforehand. To the best of our knowledge, the issue of ICL in semantic segmentation has only been addressed in [115, 114, 158, 103]. Ozdemir *et al.* [115, 114] present an ICL approach for medical imaging, modifying a standard image-level classification method [81] for segmentation and devising a method to select relevant samples from old datasets for rehearsal. Similarly, Taras et al. [158] propose a strategy for segmenting remote sensing data. On the other hand, Michieli et al. [103] tackle ICL for semantic segmentation under a specific scenario where labels for old classes are given while learning new classes, with the assumption that novel classes are never present as background in previous learning steps. These assumptions limit the applicability of their method to real-world scenarios. In this work, we present a more comprehensive formulation of the ICL problem in semantic segmentation. Unlike previous studies [115, 158, 103], our analysis is not restricted to medical [115] or remote sensing data [158] and we do not impose any constraints on how the label space should evolve across different learning steps [103]. Addionally, we provide a comprehensive experimental evaluation of state of the art ICL methods on two novel ICL benchmarks in semantic segmentation.

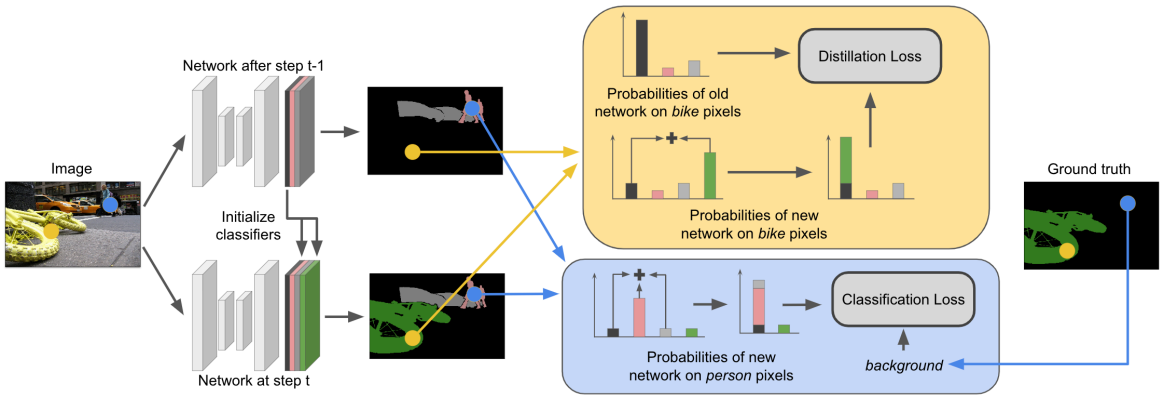


Fig. 3.2 Our method operates as follows: during each learning step t, an image is processed by both the old (top) and current (bottom) models.We use a cross-entropy loss to learn new classes (depicted by the blue block) and a distillation loss to retain previous knowledge (represented by the yellow block). To handle the semantic changes in the background, we implement the following steps: (i) initialize the new classifier using the weights of the previous background classifier (as shown on the left), (ii) compare the pixel-level background ground truth in the cross-entropy loss with the probability of having either the background class (black) or an old class (represented by the pink and grey bars), and (iii) link the background probability given by the old model in the distillation loss to the probability of having either the background or a new class (depicted by the green bar).

3.2.2 MiB: Modeling The Background

Problem Definition. As described in Sec. 2.1, we recall the ICL setting training is realized over multiple phases, called *learning steps*, and each step introduces novel categories to be learned. In other terms, during the t-th learning step, the previous label set $\mathscr{Y}^{t-1}$ is expanded with a set of new classes $\mathscr{C}^t$, yielding a new label set $\mathscr{Y}^t = \mathscr{Y}^{t-1} \cup \mathscr{C}^t$. At the learning step t, a model $f_{\theta^t} : \mathscr{X} \mapsto (\mathscr{Y}^t)^N$, with parameters θ^t, has to be updated for learning the set of new classes $\mathscr{C}^t$ using a training set $\mathscr{T}^t \subset \mathscr{X} \times (\mathscr{C}^t)^N$. In addition, to extract knowledge of the previous classes the training can rely on a frozen copy of the model trained in the previous step $t-1$, *i.e.* $f_{\theta^{t-1}} : \mathscr{X} \mapsto (\mathscr{Y}^{t-1})^N$. As in ICL for image classification, we assume that the labels set $\mathscr{C}^t$ at training step t is disjoint to previous label sets except for the special background class $\mathtt{b}$. In other words, the dataset contintains annotation only for the novel classes and the other pixels are assigned to the background class $\mathtt{b} \in \mathscr{Y}$.

Knowledge Distillation Framework. A simple method to tackle the ICL issue is to train the model f_{θ^t} on each dataset $\mathscr{T}^t$ one after the other. This is equivalent to fine-tuning the deep network parameters of f_{θ^t} on $\mathscr{T}^t$ using the parameters θ^{t-1} from the previous stage. While straightforward, this method causes catastrophic forgetting, as there are no samples from the previously seen classes in $\mathscr{T}^t$, leading to a bias towards the novel categories $\mathscr{C}^t$

at the expense of the classes from previous sets. To overcome this in the context of image-level classification ICL, a common approach is to couple the supervised loss on $\mathscr{T}^t$ with a regularization term, either considering the significance of each parameter for previous tasks [69, 144], or distilling knowledge using the predictions of the old model $f_{\theta^{t-1}}$ [81, 132, 20]. Our work takes inspiration from the latter solution and minimizes the following loss function:

$$\mathscr{L}(\theta^t) = \frac{1}{|\mathscr{T}^t|} \sum_{(x,y)\in\mathscr{T}^t} (\ell_{ce}(x,y) + \lambda \ell_{kd}(x)) \tag{3.1}$$

where ℓ_{ce} is the standard supervised loss (*e.g.* cross-entropy loss), ℓ_{kd} is the distillation loss, and $\lambda > 0$ is an hyperparameter that balances the two terms.

We recall that in semantic segmentation we have the set of new classes $\mathscr{C}^t$ and old classes $\mathscr{Y}^{t-1}$ that share the void/background class $\mathtt{b}$. However, the distribution of the background class changes between incremental steps, as its annotations in $\mathscr{T}^t$ may refer to every class not in $\mathscr{C}^t$, i.e., classes that could belong to $\mathscr{Y}^{t-1}$ or to future unseen classes $\mathscr{C}^u$ with $u > t$. To account for this semantic shift in the background class distribution, in the following we revisit the standard choice for the general objective function defined in Eq. (3.1).

Revisiting Cross-Entropy Loss. A pixel-level cross-entropy loss function is a standard choice as ℓ_{ce} in Eq.(3.1). It is calculated over all image pixels as follows:

$$\ell_{ce}(x,y) = -\frac{1}{N} \sum_{i\in\mathscr{I}} \log q_x^t(i,y_i)\,, \tag{3.2}$$

where $y_i \in \mathscr{Y}^t$ is the ground truth label associated to pixel i and $q_x^t(i,c)$ represents the probability of class c in pixel i calculated by the model f_{θ^t} for the image x.

However, using this cross-entropy loss function in Eq.(3.2) with the training set $\mathscr{T}^t$, that only contains information about the novel classes in $\mathscr{C}^t$, can result in even more severe catastrophic forgetting problem. This is because the background class in $\mathscr{T}^t$ could also contain pixels associated with the previously seen classes in $\mathscr{Y}^{t-1}$. To address this issue, we modify the cross-entropy loss function in Eq.(3.2) as follows:

$$\ell_{ce}(x,y) = -\frac{1}{N} \sum_{i\in\mathscr{I}} \log \tilde{q}_x^t(i,y_i)\,, \tag{3.3}$$

where:

$$\tilde{q}_x^t(i,c) = \begin{cases} q_x^t(i,c) & \text{if } c \neq \mathtt{b} \\ \sum_{k\in\mathscr{Y}^{t-1}} q_x^t(i,k) & \text{if } c = \mathtt{b}\,. \end{cases} \tag{3.4}$$

With this modified cross-entropy loss function, our model can be updated to predict the new classes while also accounting for the uncertainty over the content of the background class. The comparison between the ground truth and its probabilities is no longer direct, but with the probability of having either an old class or the background, as predicted by the current model f_{θ^t} (Eq.(3.4)). The procedure is depicted in the blue block of Fig. 3.2. We note that simply ignoring the background pixels within the cross-entropy loss is sub-optimal, as it does not allow for adapting the background classifier to its semantic shift and for utilizing the information about old classes in the new images.

Revisiting Distillation Loss. Distillation loss [57] is a common strategy in incremental learning to transfer information from the old model $f_{\theta^{t-1}}$ into the new one in order to prevent catastrophic forgetting. A usual choice for the distillation loss ℓ_{kd} is:

$$\ell_{kd}(x) = -\frac{1}{N} \sum_{i \in \mathscr{I}} \sum_{c \in \mathscr{Y}^{t-1}} q_x^{t-1}(i,c) \log \hat{q}_x^t(i,c)\,, \tag{3.5}$$

where $\hat{q}_x^t(i,c)$ refers to the re-normalized probability of class c for pixel i given by f_{θ^t}. Re-normalized is performed across all the old classes in $\mathscr{Y}^{t-1}$, *i.e.* :

$$\hat{q}_x^t(i,c) = \begin{cases} 0 & \text{if } c \in \mathscr{C}^t \setminus \{\mathtt{b}\} \\ q_x^t(i,c) / \sum_{k \in \mathscr{Y}^{t-1}} q_x^t(i,k) & \text{if } c \in \mathscr{Y}^{t-1}\,. \end{cases} \tag{3.6}$$

The idea behind ℓ_{kd} is to encourage f_{θ^t} to generate activations similar to those generated by $f_{\theta^{t-1}}$. This helps regulate the training process, so that the parameters θ^t remain anchored to the solution obtained for the previous classes, that is, θ^{t-1}.

The loss defined in Eq.(3.5) has been used in various forms in different contexts, from incremental-task learning [81] and incrental-class learning in image classification [132, 20] to more complex scenarios such as detection [145] and segmentation [103]. Despite its effectiveness, it has a major drawback in semantic segmentation: it completely ignores the fact that the background class is shared across different learning steps. With Eq.(3.3) we handled the semantic shift of the background regarding old classes (i.e. $\mathtt{b} \in \mathscr{T}^t$ includes pixels from $\mathscr{Y}^{t-1}$). Differently, we use the distillation loss to handle the background-shift issue for the future classes: the background in a previous step s, $s < t$, might have included pixels of the current classes in $\mathscr{C}^t$.

Considering the above, the probabilities assigned to a pixel as background by the old model $f_{\theta^{t-1}}$ and the current model f_{θ^t} do not have the same semantic meaning. More importantly, $f_{\theta^{t-1}}$ might predict pixels of classes in $\mathscr{C}^t$ as background, which we are trying

to learn. This aspect, unique to the segmentation task and not present in previous incremental learning methods, must be taken into account when distilling the old model into the new one. To address this issue, we redefine the distillation loss defining the probabilities $\hat{q}_x^t(i,c)$ in Eq.(3.6) as follows:

$$\hat{q}_x^t(i,c) = \begin{cases} q_x^t(i,c) & \text{if } c \neq \texttt{b} \\ \sum_{k \in \mathscr{C}^t} q_x^t(i,k) & \text{if } c = \texttt{b}\,. \end{cases} \tag{3.7}$$

Like Eq.(3.5), we still compare the probabilities of a pixel belonging to seen classes as assigned by the old model with the current parameters θ^t. However, unlike classical distillation, in Eq.(3.7) the probabilities obtained with the current model are not altered and are normalized across the entire label space $\mathscr{Y}^t$, instead of with respect to the subset $\mathscr{Y}^{t-1}$ (as in Eq.(3.6)). Most 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} (as shown in the yellow block in Fig. 3.2).

The novel distillation loss in Eq.(3.7) have several advantages with respect to Eq.(3.6) or other simple choices (*e.g.* excluding b from Eq.(3.6)). In our approach, the probabilities assigned to a pixel by the old model $f_{\theta^{t-1}}$ are compared with their counterparts in the current model f_{θ^t}, while considering the entire label space $\mathscr{Y}^t$, not just the seen classes $\mathscr{Y}^{t-1}$. Additionally, the probability of the background class assigned by $f_{\theta^{t-1}}$ is compared with the sum of probabilities of either a new class or the background as assigned by f_{θ^t}. This allows for distillation of full information from the old model, without constraining pixels or classes, and also propagates uncertainty of the semantic content of the background without penalizing new classes being learned.

Classifiers' Parameters Initialization. As mentioned before, the background class, represented by b, serves as a placeholder for pixels that belong to an unknown object class. At each learning step t, the new classes in $\mathscr{C}^t$ are unknown to the previous classifier $f_{\theta^{t-1}}$. This means that unless the appearance of a class in $\mathscr{C}^t$ closely resembles a class in $\mathscr{Y}^{t-1}$, $f_{\theta^{t-1}}$ will likely assign pixels in $\mathscr{C}^t$ to b. Given this initial bias in the predictions of f_{θ^t} on pixels in $\mathscr{C}^t$, randomly initializing the classifiers for the new classes is sub-optimal. This is because it creates a mismatch between the features extracted by the model, which are aligned with the background classifier, and the randomly assigned parameters of the new classifier. As a result, the network may initially assign high probabilities for pixels in $\mathscr{C}^t$ to b, potentially causing instability in the training process.

To tackle this challenge, we propose a method to initialize the parameters of the classifier for novel classes in a way that distributes the probability of background uniformly among

the classes in $\mathscr{C}^t$. For an image x and a pixel i, we set $q_x^t(i,c) = q_x^{t-1}(i,\mathtt{b})/|\mathscr{C}^t|; \forall c \in \mathscr{C}^t$, where $|\mathscr{C}^t|$ is the number of new classes (notice that $\mathtt{b} \in \mathscr{C}^t$). Without loss of generality, we consider a standard fully connected classifier. The parameters for class c at step t are denoted as $\omega_c^t, \beta_c^t \in \theta^t$, with ω and β representing the weights and bias, respectively. The initialization of ω_c^t, β_c^t can be done as follows:

$$\omega_c^t = \begin{cases} \omega_{\mathtt{b}}^{t-1} & \text{if } c \in \mathscr{C}^t \\ \omega_c^{t-1} & \text{otherwise} \end{cases} \tag{3.8}$$

$$\beta_c^t = \begin{cases} \beta_{\mathtt{b}}^{t-1} - \log(|\mathscr{C}^t|) & \text{if } c \in \mathscr{C}^t \\ \beta_c^{t-1} & \text{otherwise} \end{cases} \tag{3.9}$$

where $\{\omega_{\mathtt{b}}^{t-1}, \beta_{\mathtt{b}}^{t-1}\}$ are the weights and bias of the background class in the classifier of the previous learning step. It is simple to see that the initialization in Eq.(3.8) and (3.9) results in $q_x^t(i,c) = q_x^{t-1}(i,\mathtt{b})/|\mathscr{C}^t| \; \forall c \in \mathscr{C}^t$, as $q_x^t(i,c) \propto \exp(\omega_{\mathtt{b}}^t \cdot x + \beta_{\mathtt{b}}^t)$.

We will demonstrate through our experimental analysis that this simple initialization approach has positive impacts on both the stability of the model during learning and the final outcomes. This is because it reduces the burden of supervision imposed by Eq.(3.3) during the learning of new classes and aligns with the principles behind our distillation loss (Eq.(3.7)).

3.2.3 Experiments

ICL Baselines. In this study, our method is compared to standard ICL baselines, which were originally designed for classification tasks, on the segmentation task by treating it as a pixel-level classification problem. The results of six different methods, including three structural-based methods and three functional-based approaches, are reported.

In the structural-based category, Elastic Weight Consolidation (EWC) [69], Path Integral (PI) [194], and Riemannian Walks (RW) [24] were selected. These methods use different techniques to calculate the importance of each parameter for old classes: EWC employs the empirical Fisher matrix, PI uses the learning trajectory, and RW combines EWC and PI in a unique model. EWC was chosen as it is a standard baseline used in [145]. PI and RW were selected because they are simple applications of the same principle. These methods operate at the parameter level, and to adapt them to the segmentation task, we kept the loss in the output space unchanged (i.e., standard cross-entropy across the entire segmentation mask)

and computed the importance of the parameters by considering their effect on learning in previous training steps.

In the functional-based category, Learning without forgetting (LwF) [81], LwF multi-class (LwF-MC) [132], and the segmentation method of [103] (ILT) were chosen. LwF refers to the original distillation-based objective implemented in Eq.(3.1) with basic cross-entropy and distillation losses, which is the same as [81] except that distillation and cross-entropy share the same label space and classifier. LwF-MC is the single-head version adapted from [81] as described in [132]. It employs multiple binary classifiers, with the target labels defined using the ground truth for novel classes (i.e., $\mathscr{C}^t$) and the probabilities provided by the old model for the old ones (i.e., $\mathscr{Y}^{t-1}$). Since the background class is both in $\mathscr{C}^t$ and $\mathscr{Y}^{t-1}$, LwF-MC is implemented by a weighted combination of two binary cross-entropy losses, one for the ground truth and the other for the probabilities provided by $f_{\theta^{t-1}}$. Finally, ILT [103] is the only method specifically designed for ICL in semantic segmentation. It uses distillation loss in the output space, similar to our adapted version of LwF [81], and/or distillation loss in the feature space attached to the network decoder's output. In this study, we use the variant where both losses are employed. We do not compare our method to methods that use rehearsal (e.g., [132]), as they violate the standard ICL assumption of the unavailability of old data, as done in [145].

We also include two other baselines in all tables, simple fine-tuning (FT) on each $\mathscr{T}^t$ (as described in Eq.(3.2)) and training on all classes offline (Joint). The latter can be considered an upper bound. Our method is referred to as MiB (**M**odel**i**ng the **B**ackground) in the tables. The results are presented as the mean Intersection-over-Union (mIoU) in percentage, averaged over old, new and all classes, after the final incremental learning step.

Implementation Details. For all the methods, we utilize the Deeplab-v3 [27] architecture with a ResNet-101 [55] backbone and an output stride of 16. To address memory constraints, which are a crucial issue in semantic segmentation, we implement in-place activated batch normalization [138]. The backbone of the network is initialized using the ImageNet pretrained model given by [138]. We follow the training protocol outlined in [27] and use Stochastic Gradient Descent (SGD) with the same learning rate policy, momentum, and weight decay. Our initial learning rate is set to 10^{-2} for the first learning step and 10^{-3} for subsequent steps, as described in [145]. We train the network with a batch size of 24 for 30 epochs on the Pascal-VOC 2012 dataset and 60 epochs on the ADE20K dataset for each learning step. The data augmentation procedure is the same as that described in [27] and the images are cropped to 512×512 during both training and testing. To set the hyperparameters for each method, we use the incremental learning protocol defined in [34], and we utilize 20% of the

Table 3.1 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 [194]	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 [69]	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 [24]	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 [81]	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 [132]	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 [103]	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

training set as a validation set. The final results are reported on the standard validation set of the datasets.

Pascal-VOC 2012

We define two experimental settings for PASCAL-VOC [43] based on the way images are sampled to create the incremental datasets. The first experimental setting, referred to as the *disjoint* setup, is based on the work in [103]. In this setup, each learning step consists of a unique set of images, with pixels belonging to classes seen in either the current or previous learning steps. However, in contrast to [103], only the labels for pixels of novel classes are assumed to be available at each step, while the labels for the old classes are considered as background in the ground truth. The second experimental setting, referred to as the *overlapped* setup, is based on the incremental learning for object detection setting [145]. In this setup, each step contains all the images that have at least one pixel of a novel class, with only the latter annotated. Unlike the disjoint setup, images may contain pixels of classes that will be learned in the future, but they are labeled as background. This setup is more realistic as it does not make any assumptions about the objects present in the images.

In line with previous works [145, 103], we perform three experiments on 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 determine the content of each learning step.

Addition of one class *(19-1)*. In this experiment, we conduct two learning steps. The first step involves observing the first 19 classes, while the second step involves learning the *tv-monitor* class. The results are reported in Table 3.1. Without the use of any regularization strategy, the performance on past classes significantly decreases. The FT method performs poorly, completely forgetting the first 19 classes. Surprisingly, using PI as a regularization strategy does not result in any benefits, while EWC and RW improve performance by nearly

15%. On the other hand, structural-based strategies are outperformed by functional-based methods, with LwF, LwF-MC, and ILT performing better by a large margin. This confirms the effectiveness of functional-based methods in preventing catastrophic forgetting. Our method, which improves on standard ICL baselines, is particularly impressive in terms of new classes, resulting in a 11% improvement in mIoU, without any forgetting of old classes. When compared to LwF, our method provides an average improvement of around 15%, demonstrating the importance of modeling the background in ICL for semantic segmentation. These results are consistent across both the "disjoint" and "overlapped" scenarios.

Single-step addition of five classes (*15-5*). The following classes are added after the first training set in this experiment: *plant, sheep, sofa, train, tv-monitor*. The results are presented in Table 3.1. As in the 19-1 setting, FT and PI show a significant decline in performance, while functional-based strategies (LwF, LwF-MC, ILT) outperform EWC and RW significantly. Our method, on the other hand, achieved the best results, approaching the upper bound of joint training. In the *disjoint* setup, our method outperforms the best baseline by 4.6% for the old classes, 2% for the novel classes, and 4% in all classes. These gaps are even larger in the *overlapped* setting, where our method surpasses the baselines by almost 10% in all cases, highlighting its ability to exploit information in the background class.

Multi-step addition of five classes (*15-1*). The results in this setting, where the last 5 classes are learned one by one, are presented in Table 3.1. This setting proves to be difficult for existing methods, as they fail to perform well with a score below 7% on both old and new classes. FT and structural-based techniques cannot retain prior knowledge and heavily favor new classes, resulting in poor performance on the first 15 classes. The functional-based methods also experience a significant decline in performance, losing over 50% of their score from the single to multi-step scenario. However, our method still manages to perform well, outperforming all baselines by a wide margin in both old (46.2% in the disjoint and 35.1% in the overlapped scenario) and new (nearly 13% in both setups) classes. The results demonstrate that the overlapped scenario is particularly challenging, as it does not limit which classes are present in the background, leading to an overall performance drop of 11% on all classes.

Ablation Study. In Table 3.2, we present an extensive evaluation of our contributions in the *overlapped* setup. Our analysis begins with the baseline LwF [81] that utilizes standard cross-entropy and distillation losses. Firstly, we incorporate our modified cross-entropy (*CE*) into the baseline, which enhances the capability of retaining old knowledge in all scenarios without causing harm (*15-1*) or even improving (*19-1*, *15-5*) the performance on the novel classes. Secondly, we incorporate our distillation loss (*KD*) into the model, resulting in an

Table 3.2 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 [81]	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**

Table 3.3 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 [81]	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 [132]	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 [103]	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

improvement in performance for both old and new classes. The improvement in old classes is significant, especially in the 15-1 scenario (*i.e.* 22.8%). The improvement in the novel classes is consistent and particularly pronounced in the 15-5 scenario (7%). This demonstrates the unique aspect of our *KD* as it not only preserves old knowledge but also contributes to the learning of new classes. Lastly, we add our classifiers' initialization strategy (*init*) which leads to an improvement in all scenarios, especially in the novel classes. It doubles the performance in the *19-1* scenario (22.1% vs 11.9%) and triples it in the *15-1* scenario (4.5% vs 13.5%), emphasizing the significance of considering the background shift during the initialization stage for better learning of the new classes.

ADE20K

We create incremental datasets $\mathscr{T}^t$ by dividing the entire dataset into separate image sets with a minimum of 50 images per class in $\mathscr{C}^t$ that have labeled pixels. Each $\mathscr{T}^t$ provides annotations only for the classes in $\mathscr{C}^t$ while the remaining classes, both old and future, are considered as background in the ground truth. In Table 3.3, we present the mean IoU obtained by averaging the results from two different class orders: one proposed by [202] and a random order. In this experiment, we compare our approach with functional-based methods only (LwF, LwF-MC, and ILT) due to their lower performance compared to structural-based methods.

Single-step addition of 50 classes (*100-50*). In the first experiment, we first train the network with 100 classes, then add the remaining 50 classes all at once. The results in Table

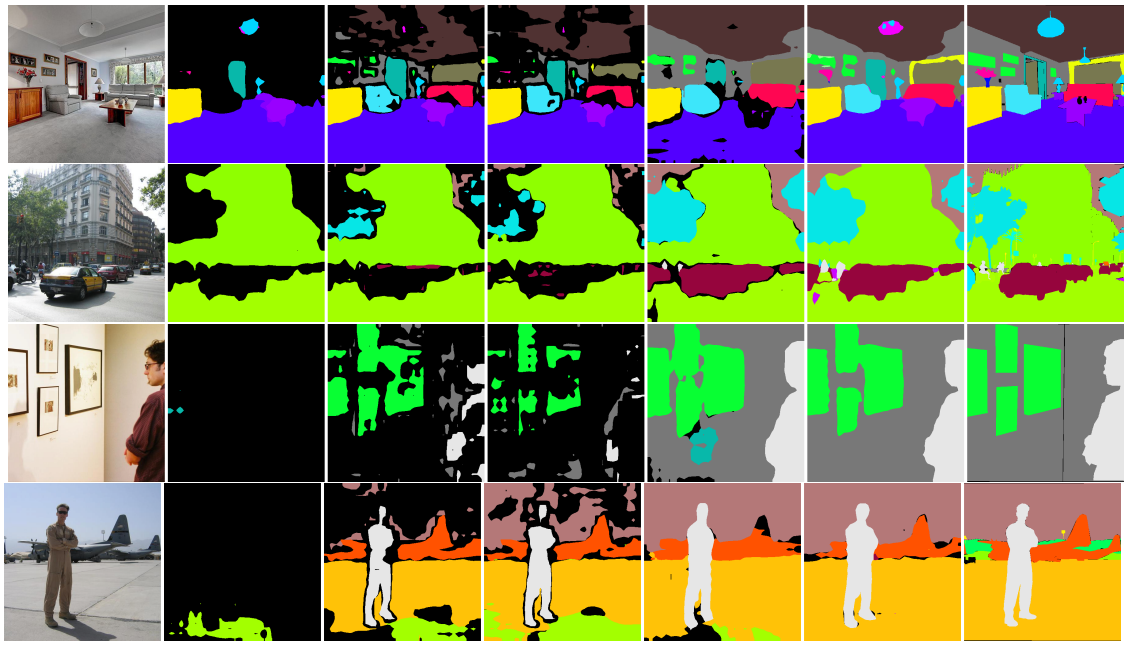


Fig. 3.3 Qualitative results on the *100-50* setting of the ADE20K dataset using different incremental methods. The image demonstrates the superiority of our approach on both new (*e.g. building*, *floor*, *table*) and old (*e.g. car*, *wall*, *person*) classes. From left to right: image, FT, LwF [81], ILT [103], LwF-MC [132], our method, and the ground-truth. Best viewed in color.

3.3 show that the fine-tuning (FT) strategy performs poorly in large-scale settings, causing complete forgetting of old knowledge. Using a distillation strategy, such as LwF, ILT, and LwF-MC, reduces catastrophic forgetting. Among these methods, LwF-MC obtains the highest score of 34.2% on past classes, while LwF performs the best on new classes with a score of 18.9% higher than LwF-MC and 6.6% higher than ILT. Our method outperforms all others, showing improvements on both past and new classes. Our results are close to the upper bound of joint training, especially for new classes, with a gap of only 0.3%. In Figure 3.3, we present qualitative results that demonstrate the superiority of our method compared to the baselines.

Multi-step addition of 50 classes (*100-10*). In the second experiment, we evaluate the performance of our method in multiple incremental steps, where the network starts with 100 classes and the remaining 50 classes are added in increments of 10 classes. The results, reported in Table 3.3, show that the FT, LwF and ILT methods have poor performance due to catastrophic forgetting. LwF-MC shows a better ability to retain knowledge of previous classes, but at the cost of reduced performance on new classes. Our method, however, achieves the best balance between learning new classes and preserving previous knowledge, outperforming LwF-MC by 11.6% considering all classes.

Three steps of 50 classes (*50-50*). Finally, in Table 3.3, we examine the performance in three consecutive stages of 50 classes. The previous incremental learning methods exhibit different balances between learning new classes and avoiding forgetting old knowledge. LwF and ILT achieve good scores on new classes, but they forget old knowledge. Conversely, LwF-MC retains knowledge of the first 50 classes but fails to learn new ones. Our method surpasses all baselines by a substantial margin, with a difference of 11.9% compared to the best-performing baseline and attains the highest mIoU in each stage. Notably, the largest gap is in the intermediate stage, where classes must be learned incrementally and retained from being forgotten in the following learning stage.

3.3 Incremental Learning in Object Detection

In the prior section, we explored incremental learning in semantic segmentation, which presents an additional challenge in the form of a shift in the background semantic. Here, we will analyze a similar issue in object detection. According to the incremental learning in object detection (ILOD) definition in [145], only objects belonging to new classes are annotated, while the rest (belonging to either old or future classes) are ignored, resulting in missing annotations (see Fig. 3.4).

Research in ILOD has focused on introducing regularizations to prevent catastrophic forgetting, however, the effect of missing annotations has not been taken into account. Regions that are not annotated are usually treated as background areas and the model assigns them to a specific *background* class. This can lead to objects that are not annotated being associated with the background, which can worsen forgetting in already seen classes and make it more difficult to train for future classes.

In order to address this issue, we propose MMA, which **M**odels the **M**issing **A**nnotations, as an adaptation of the method presented in Sec. 3.2 and a revisitation of the common distillation framework in ILOD [145, 120, 203]. This approach allows the model to predict either an old class or the background on any region not associated with an annotation on the classification loss, thus reducing the risk of catastrophic forgetting. Additionally, the distillation loss is modified to match the teacher model's background probability with the probability of having either a new class or the background, allowing for easier learning of new classes. To evaluate the effectiveness of our method, experiments are conducted on the Pascal-VOC dataset [42] for a variety of single-step and multi-step tasks. Results show that our method outperforms the current state-of-the-art without using any image from previous training steps. Moreover, by adding an additional knowledge distillation term to our framework, we can extend it to the task of instance segmentation. Experiments on the Pascal-VOC dataset [43] demonstrate that our method outperforms the other baselines.

To summarize, this section contributions are: (1) the identification of the peculiar issue of missing annotations in incremental learning for object detection, which was overlooked by previous works, (2) the proposal of a novel method, MMA, that revisits the standard knowledge distillation framework and outperforms previous methods on multiple ILOD settings, and additionally (3) an extention of MMA to instance segmentation, where it exceeds all other baselines. The code to replicate experiments in this section can be found at https://github.com/fcdl94/MMA.

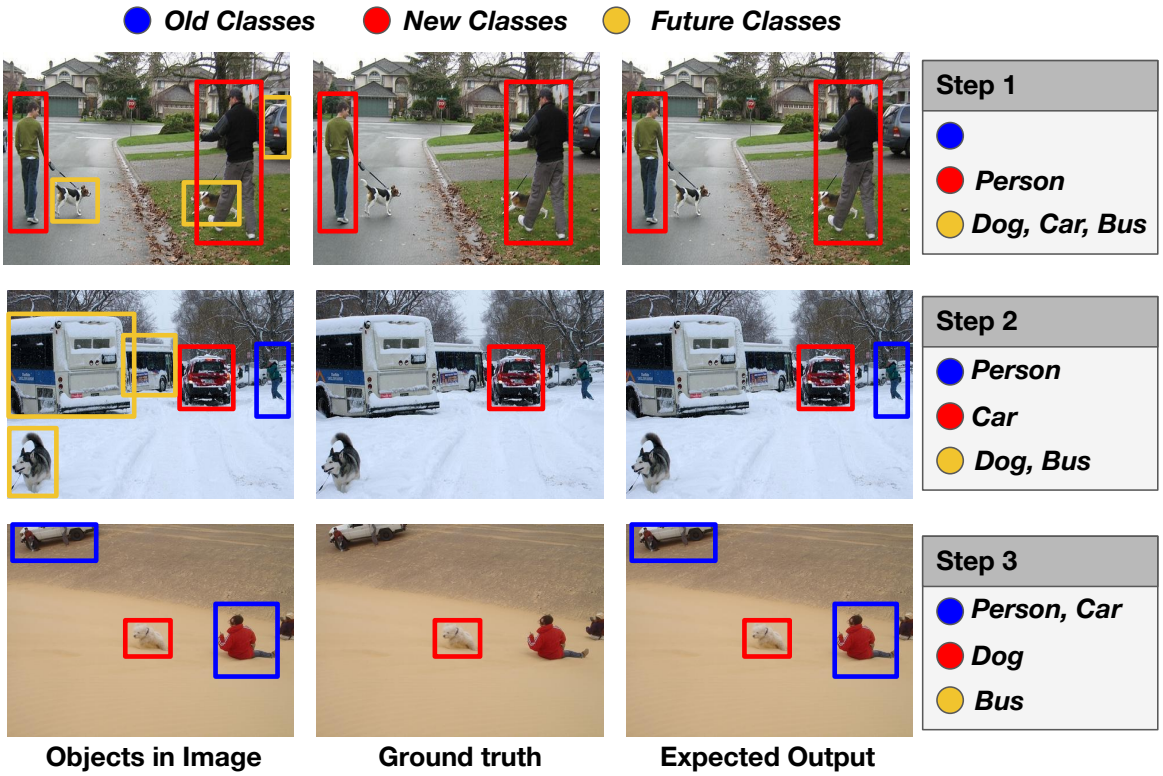


Fig. 3.4 The figure depicts the missing annotation issue in different learning steps in object detection. At the training step t, annotations are only provided for newly added classes (represented with red boxes). All other objects, both those from previous time steps (represented with blue boxes) and those from future time steps (yellow boxes) are not annotated.

3.3.1 Related Works

Object Detection architectures can be broadly classified into two categories: one-stage detectors [133, 162, 89, 19, 153, 204] and two-stage detectors [49, 48, 135, 56, 85]. The two-stage detectors generally provide better performance but are less efficient. They involve two steps: first by extracting regions of interest (RoIs) either through a neural network (e.g., Faster R-CNN [135]) or an external region proposer (e.g., Fast R-CN [48]), and then by using a multi-layer perceptron on the RoIs to obtain the final classification and bounding box regression. On the other hand, one-stage detectors directly predict the final output, without the need to predict RoIs. Although these two architecture groups are powerful in an offline setting, they are not suitable for adding new classes incrementally over time. In this work, we focus on enabling two-stage methods, specifically Faster R-CNN, to learn new categories over time without losing previous knowledge in the absence of original data.

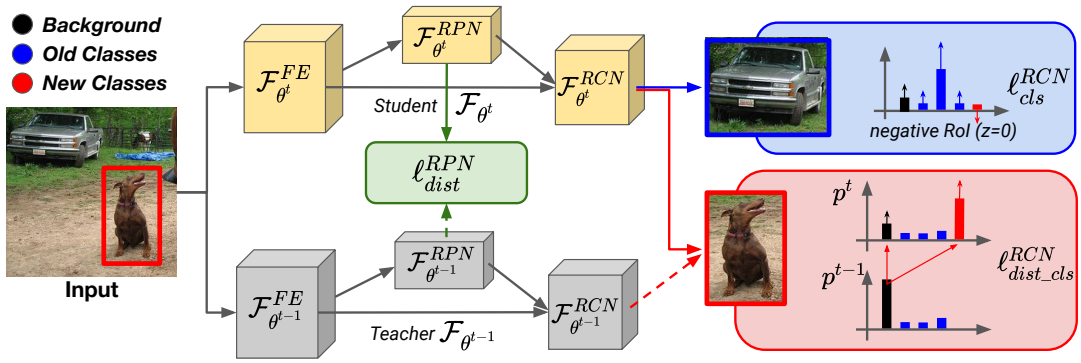


Fig. 3.5 Overview of MMA, highlighting its contributions. Given an image, it is forwarded on the student (top) and teacher (bottom) models. The blue box illustrates the behavior of revised cross entropy loss on a negative RoI (*i.e.* RoI without annotations): the model maximizes the probability of having either the background or an old class. In the red box, we show the effect of the distillation loss on the classification output for a new class region: it associates the teacher background with either the student background or a new class. Lastly, in green, it is reported the RPN distillation loss.

Incremental Learning in Object Detection. In recent years, there has been growing interest in incremental learning for object detection. A landmark study in this field is [145], which presents a framework that relies on two-stage object detectors and performs knowledge distillation on the output of Fast R-CNN [48]. Several methods have since been proposed that build upon this framework, extending it to the Faster R-CNN [135] architecture. These methods incorporate distillation terms into the intermediate feature maps [188, 120, 25, 90, 203] and aim to prevent forgetting in the region proposal network [120, 25, 53, 203]. [203] introduced a pseudo-positive-aware sampling algorithm to distinguish regions belonging to old classes and avoid classifying them as background regions. However, this approach only provides a partial solution as it does not account for missing annotations or model confidence. Some works, such as [52, 67, 70, 1], focus on rehearsal techniques to preserve old task knowledge, either by replaying intermediate features [1] or images [78, 52, 52]. [88] proposes a parameter isolated method based on EWC [69] for object detection. Additionally, a few studies have explored incremental learning using one-stage architectures [78, 121, 122]. In this study, we propose a distillation framework for two-stage architectures that explicitly models missing annotations for objects not included in the current training step.

3.3.2 MMA: Modeling the Missing Annotations

The objective of object detection is to train a model that can locate and identify objects within an image, represented by a rectangular box and a class label. The focus of this work is on the R-CNN [48, 135, 56] family of detection models, denoted as f_θ, with parameters

θ. The detection model is comprised of three components: a feature extractor f_θ^{FE}, a region proposal network (RPN) f_θ^{RPN}, and a classification head f_θ^{RCN}. Given an image x, the feature extractor generates a dense feature map, which is then passed to the RPN. The RPN's goal is to produce a set of K rectangular regions of interest (RoIs) each with a binary objectness score. The K RoIs are applied to the feature map and classified by the classification head, which outputs the class probabilities $p \in \mathbb{R}^{|\mathscr{Y}|+1}$ for each RoI, where $\mathscr{Y}$ is the set of classes, as well as rectangular boxes $r \in \mathbb{R}^{4|\mathscr{Y}|}$ corresponding to each class. It's important to note that the classifier also outputs a class score for the background to signify the absence of objects in the RoI.

In Incremental Learning for Object Detection (ILOD), the training is carried out in multiple *learning steps*, each step adding a new set of classes to be detected. In the t-th training step, a detection model f_{θ^t} is trained to learn the classes $\mathscr{C}^t$ using a training set $\mathscr{T}^t$. It's important to note that although an image in the training set $\mathscr{T}^t$ may contain multiple objects of various classes, only annotations for classes in $\mathscr{C}^t$ are provided following the ILOD protocol [145]. Additionally, the old training sets are not available at training step t. After the t-th step, the model f_{θ^t} is expected to predict for all classes seen so far, meaning its output should take into account the classes in $\mathscr{Y}^t = \cup_{t'=1}^{t} \mathscr{C}^{t'}$. We note that, following the standard practices, $\mathscr{C}^i \cap \mathscr{C}^j = \emptyset$ for any $i \neq j$.

In the standard Faster R-CNN training process as described in [135], a multi-task loss is minimized, as given by equation 3.10:

$$\ell_{faster} = \ell_{cls}^{RPN} + \ell_{reg}^{RPN} + \ell_{cls}^{RCN} + \ell_{reg}^{RCN}. \tag{3.10}$$

The first two terms in the loss, ℓ_{cls}^{RPN} and ℓ_{reg}^{RPN}, are the classification and regression loss on the Region Proposal Network (RPN), while the latter two terms, ℓ_{cls}^{RCN} and ℓ_{reg}^{RCN}, are applied on the output of the classification head [48, 135]. For more details on the training of Faster R-CNN, please refer to [48, 135]. However, despite its effectiveness, Faster R-CNN is not well-suited for updating its weights to learn new classes. Fine-tuning the model using the loss function given in equation 3.10 on a new training set $\mathscr{T}^t$ causes the model to forget the information it has previously learned, leading to catastrophic forgetting [102].

Previous studies [145, 120, 203, 53, 25] have suggested using knowledge distillation [57, 81] to mitigate the issue of forgetting in incremental learning. In this approach, a student model f_{θ^t} is trained to imitate the output of the previous model $f_{\theta^{t-1}}$. However, these studies did not address the problem of missing annotations. In incremental learning, the dataset $\mathscr{T}^t$ at time step t provides annotations only for objects in $\mathscr{C}^t$, while objects belonging to past or future classes that are present in the image are not annotated. As a result, these objects

are treated as background in the standard detection pipeline, leading to two issues: (i) if the RoI contains an object from an old class, the model is trained to predict it as background, exacerbating the forgetting issue; (ii) if the RoI contains an object from a future class, the model is trained to consider it as background, making it harder to learn new classes when they will be presented. The issue of missing annotations is similar to the background shift problem described in Sec. 3.2 for incremental learning in semantic segmentation. In the following, we adapt the equations proposed in Sec. 3.2 to address the missing annotations problem in incremental learning for object detection.

Revisiting Classification Loss. The Faster R-CNN classification loss ℓ_{cls}^{RCN} aims to ensure that the network assigns the correct class label to the Region of Interests (RoIs). To do so, the loss is computed for a set of K RoIs that have been generated by the RPN and either matched with a ground truth label (positive RoI) or classified as the background (negative RoI). The calculation of the loss can be expressed as follows:

$$\ell_{cls}^{RCN} = \frac{1}{K}\sum_{i=1}^{K} z_i \sum_{c\in\mathscr{Y}^t} \bar{y}_i^c \log(p_i^c) + (1-z_i)\log(p_i^b), \tag{3.11}$$

where z_i takes the value of 1 for positive RoIs and 0 otherwise, $\bar{y}_i$ is the one-hot encoded class label (1 for the ground truth class and 0 for others), and p_i^b represents the probability of the i-th RoI belonging to the background class.

The equation specified in Eq. (3.11) was originally created for standard object detection and as such, it does not account for the fact that only information about novel classes is available in the ground truth. This presents a problem as all other objects in the image that do not have a corresponding ground-truth association are treated as negative RoI, leading the model to learn to predict the background class for them. This becomes particularly concerning for objects belonging to old classes, as it results in severe catastrophic forgetting, causing the model to forget the correct class of the object and replace it with the background.

In order to address this issue, we modify the equation Eq. (3.11) taking inspiration from Eq. (3.4):

$$\ell_{cls}^{RCN} = \frac{1}{K}\sum_{i=1}^{K} z_i \sum_{c\in\mathscr{C}^t} \bar{y}_i^c \log(p_i^c) + (1-z_i)\log(p_i^b + \sum_{o\in\mathscr{Y}^{t-1}} p_i^o), \tag{3.12}$$

where p_i^c represents the probability of the class c for the query i, $\mathscr{C}^t$ are the classes that are newly introduced at time t, and $\mathscr{Y}^{t-1}$ refers to all the classes that were seen before the time t. By using Eq. (3.12), the model is able to learn the new classes in the positive RoIs ($z_i = 1$) while ensuring that the background class does not take precedence over the older classes.

Instead of forcing the background class to be the prediction for every negative RoI ($z_i = 0$), as Eq. (3.11) does, Eq. (3.12) allows the model to predict either the background or any old class by maximizing the sum of their probabilities. The illustration for this can be seen in the blue box of Fig. 3.5.

Revisiting Knowledge Distillation. Previous works [120, 203, 53, 25] employed two knowledge distillation loss terms in the training objective to avoid forgetting:

$$\ell = \ell_{faster} + \lambda^1 \ell_{dist}^{RCN} + \lambda^2 \ell_{dist}^{RPN}, \tag{3.13}$$

where λ^1, λ^2 are hyperparameters.

The objective of ℓ_{dist}^{RCN} is to preserve the information regarding the previously learned classes in the classification head. Previous studies [145, 120] required the student model to generate classification scores and box coordinates that are similar to the teacher model for the old classes using an L2 loss. However, these works ignored the missing annotations, meaning that the new classes have been seen previously, but since they lacked annotations, they were labeled as the background class. This would cause the teacher model to predict high background scores for the new class RoIs, making it more challenging for the student model to learn the new classes, conflicting with the classification loss. To address this, we propose a distillation loss that takes into account the missing annotations, formulated as follows:

$$\ell_{dist}^{RCN} = \frac{1}{K} \sum_{i=1}^{K} \ell_{dist\_cls}^{RCN}(i) + \ell_{smooth\_l1}(r_i^t, r_i^{t-1}), \tag{3.14}$$

$$\ell_{dist\_cls}^{RCN}(i) = \frac{1}{|\mathscr{Y}^{t-1}| + 1} (p_i^{b,t-1} \log(p_i^{b,t} + \sum_{j \in \mathscr{C}^t} p_i^{j,t}) + \sum_{c \in \mathscr{Y}^{t-1}} p_i^{c,t-1} \log(p_i^{c,t})), \tag{3.15}$$

where the terms $p_i^{k,t-1}$, r_i^{t-1} and $p_i^{k,t}$, r_i^t represent, respectively, the classification and regression output for proposal i and class k for the teacher and student model, and b represents the background class. In order to handle the classification scores, we propose modifying with respect to previous works [145, 120] the first term in equation Eq. (3.14) that takes into account the box coordinates. Similarly to Eq. (3.7), to model the missing annotations, equation Eq. (3.15) uses all the class probabilities of the student model to match those of the teacher model. The old classes $\mathscr{Y}^{t-1}$ remain unchanged between the student and teacher models, while the teacher's background $p_i^{b,t-1}$ is either associated with a new class or the student's background. By using equation Eq. (3.15), when the teacher predicts a high background probability for a RoI belonging to a new class, the student is not forced to imitate

this behavior but instead has the opportunity to solidify its new knowledge and predict the correct class. This is illustrated in the red box of figure Fig. 3.5.

On the other hand, the purpose of ℓ_{dist}^{RPN} is to prevent the RPN from forgetting. Since annotations for old classes are not available, the RPN is trained to predict a high objectness score only for RoIs belonging to new classes. To ensure that the RPN continues to predict a high objectness score for regions belonging to old classes, we utilize the loss proposed by [120]. The student is made to imitate the teacher only in regions belonging to old classes, where the teacher score is higher than the student score. For a set of A regions, ℓ_{dist}^{RPN} is calculated as:

$$\ell_{dist}^{RPN} = \frac{1}{A}\sum_{i=1}^{A} \mathbb{1}_{[s_i^t \geq s_i^{t-1}]} ||s_i^t - s_i^{t-1}|| + \mathbb{1}_{[s_i^t \geq s_i^{t-1}+\tau]} ||\omega_i^t - \omega_i^{t-1}||, \tag{3.16}$$

where s_i^t is the objectness score and ω_i^t the coordinates of $f_{\theta^t}^{RPN}$ on the i-th proposal, $||\cdot||$ is the euclidean distance, τ is a hyperparameter, and $\mathbb{1}$ is the indicator function equal to 1 if the condition in the brackets is satisfied and 0 otherwise. It is important to note that when the teacher produces an objectness score greater than the student, i.e., $s_i^t > s_i^{t-1}$, the proposal is likely to contain an old class. On the other hand, when $s_i^t \geq s_i^{t-1}$, the proposal is likely to belong to a new class. In this case, forcing the student to mimic the teacher score may result in errors that negatively impact the performance on new classes.

Extension to Instance Segmentation The purpose of instance segmentation is to generate a precise mask that identifies each object in an image at the pixel level. To achieve this, we utilize Mask R-CNN [56] which is an extension of Faster R-CNN that includes a mask head f_θ^{MASK}. This mask head generates a binary segmentation mask for each Region of Interest (RoI) with a shape of $|\mathscr{Y}| \times H \times W$, where $\mathscr{Y}$ represents the set of classes and H,W represents the resolution of the mask. The mask head is trained with an additional loss term that is combined with the multi-task loss in Eq. (3.10). The Mask R-CNN loss is defined as:

$$\ell_{mask} = \ell_{faster} + \ell_{cls}^{MASK}, \tag{3.17}$$

where ℓ_{cls}^{MASK} is the per-pixel binary cross-entropy loss between the output of f_θ^{MASK} and the binary mask of the ground truth class. For further information, please refer to [56].

In spite of the fact that the approach described in Sec. 3.3.2 already takes into account forgetting in the detection head, the application of Eq. (3.17) brings the risk of forgetting how to segment past objects while learning the new ones. Therefore, we further extend Eq. (3.13) to incorporate a knowledge distillation term in the mask head. Formally, in the

Table 3.4 mAP@0.5% results on single incremental step on Pascal-VOC 2007. Methods with † come from reimplementation. Methods with * use exemplars.

	19-1				15-5				10-10			
Method	**1-19**	**20**	**1-20**	**Avg**	**1-15**	**16-20**	**1-20**	**Avg**	**1-10**	**11-20**	**1-20**	**Avg**
Joint Training	75.3	73.6	75.2	74.4	76.8	70.4	75.2	73.6	74.7	75.7	75.2	75.2
Fine-tuning	12.0	62.8	14.5	37.4	14.2	59.2	25.4	36.7	9.5	62.5	36.0	36.0
ILOD (Fast R-CNN) [145]	68.5	62.7	68.3	65.6	68.3	58.4	65.9	63.4	63.2	63.1	63.2	63.2
ILOD (Faster R-CNN) [145] †	70.3	65.2	70.0	**67.8**	72.5	58.0	68.9	65.3	69.2	53.0	61.1	61.1
Faster ILOD [120]	68.9	61.1	68.5	65.0	71.6	56.9	67.9	64.3	69.8	54.5	62.1	62.1
Faster ILOD [120] †	70.9	64.3	70.6	67.6	73.5	55.6	69.1	64.6	71.1	52.3	61.7	61.7
PPAS [203]	70.5	53.0	69.2	61.8					63.5	60.0	61.8	61.8
MVC [188]	70.2	60.6	69.7	65.4	69.4	57.9	66.5	63.7	66.2	66.0	66.1	66.1
OREO* [67]	69.4	60.1	68.9	64.7	71.8	58.7	68.5	65.2	60.4	68.8	64.6	64.6
OW-DETR* [52]	70.2	62.0	69.8	66.1	72.2	59.8	69.1	66.0	63.5	67.9	65.7	65.7
ILOD-Meta* [70]	70.9	57.6	70.2	64.2	71.7	55.9	67.8	63.8	68.4	64.3	66.3	66.3
MMA	71.1	63.4	**70.7**	67.2	73.0	60.5	**69.9**	**66.7**	69.3	63.9	**66.6**	**66.6**

case of instance segmentation, we use the following training objective:

$$\ell = \ell_{mask} + \lambda_1 \ell_{dist}^{RCN} + \lambda_2 \ell_{dist}^{RPN} + \lambda_3 \ell_{dist}^{MASK}, \tag{3.18}$$

where λ_1, λ_2, λ_3 are hyperparameters.

The objective of ℓ_{dist}^{MASK} is to keep the segmentation masks for the old classes close to the output of the teacher model. To do so, we employ a per-pixel binary cross-entropy loss between the teacher model masks and the student ones. Specifically, for each pixel i and class c in the set of old classes $\mathscr{Y}^{t-1}$, we compute a binary cross-entropy loss between the teacher model mask $m_{c,i}^{t-1} = f_{\theta^{t-1}}^{MASK}$ and the student one $m_{c,i}^{t} = f_{\theta^{t}}^{MASK}$. Formally, we have:

$$\ell_{dist}^{MASK} = \frac{1}{N|\mathscr{Y}^{t-1}|} \sum_{i=1}^{N} \sum_{c \in \mathscr{Y}^{t-1}} m_{c,i}^{t-1} \log(m_{c,i}^{t}) + (1 - m_{c,i}^{t-1}) \log(1 - m_{c,i}^{t}), \tag{3.19}$$

where N is the number of pixels in the image, i.e., $N = H \times W$. We note that this loss is only computed for the segmentation masks belonging to the old classes, while the masks belonging to the new ones are not considered.

3.3.3 Experiments

Experimental Protocol. We evaluate our MMA approach on the Pascal-VOC dataset. Specifically, we use the PASCAL-VOC 2007 [42] for object detection consisting of 5K images with bounding box annotations for training and 5K for testing. For instance segmentation, we employ the Pascal-VOC 2012 [43, 54] dataset which also reports the instance

Table 3.5 mAP@0.5% results on multi incremental steps on Pascal-VOC 2007. Methods with † come from reimplementation.

	10-5				10-2				15-1				10-1			
Method	**1-10**	**11-20**	**1-20**	**Avg-S**	**1-10**	**11-20**	**1-20**	**Avg-S**	**1-15**	**16-20**	**1-20**	**Avg-S**	**1-10**	**11-20**	**1-20**	**Avg-S**
Joint Training	74.7	75.7	75.2	75.2	74.7	75.7	75.2	75.2	76.8	70.4	75.2	73.5	74.7	75.7	75.2	75.2
Fine-tuning	6.6	28.3	17.4	21.8	5.2	12.3	8.8	16.7	0.0	8.0	2.4	6.7	0.0	4.6	2.3	8.6
ILOD (Faster R-CNN) [145] †	67.2	59.4	63.3	65.2	62.1	49.8	55.9	62.2	65.6	47.6	60.2	65.8	52.9	41.5	47.2	59.1
Faster ILOD [120] †	68.3	57.9	63.1	65.5	64.2	48.6	56.4	62.8	66.9	44.5	61.3	67.1	53.5	41.0	47.3	60.4
MMA	66.7	61.8	**64.2**	**67.3**	65.0	53.1	**59.1**	**63.8**	68.3	54.3	**64.1**	**67.5**	59.2	48.3	**53.8**	**62.4**

segmentation annotations. We use the standard instance segmentation split of Pascal-VOC 2012, using 8498 images for training and 2857 for evaluation. Following [145], we implement the following experimental protocol for both object detection and instance segmentation: each training step contains all the images that have at least one bounding box of a novel class. It is important to note that at each training step, labels for bounding boxes of novel classes are assumed to be available, while all the other objects that appear in the image, either belonging to past or future classes, are not annotated. This is a realistic setup since it does not make any assumption on the objects present in the images and reduces the amount of annotation required in each incremental step.

Implementation Details. For object detection, we followed the same approach as in previous works [120, 203, 188, 52, 67, 70], using the Faster R-CNN architecture with a ResNet-50 backbone. Similarly, for instance segmentation, we employed the Mask R-CNN [56] architecture with a ResNet-50 backbone. Both backbones were initialized using the ImageNet pretrained model [35]. We followed the same training protocol as in [145, 120], but we increased the batch size from 1 to 4 in order to reduce the time required for training, scaling accordingly the learning rate and number of iterations. In particular, for object detection we trained the network with SGD, weight decay 10^{-4} and momentum 0.9. We used an initial learning rate of $4 \cdot 10^{-3}$ for the first learning step and $4 \cdot 10^{-4}$ in the subsequent steps. We performed 10K iterations when adding 5 or 10 classes, while we trained for 2.5K when learning only one or two classes. We applied the same data augmentation as in [145, 120]. We set λ_2 equal to 0.1, 0.5, and 1 when adding 10 classes, 5, and 1 or 2 classes, respectively. λ_1, λ_3 were set to 1.

Object Detection Results

We evaluate our method by considering experimental settings with a different number of classes in one or multiple training steps, as done by previous works [188, 203, 70, 145, 120]. Specifically, we report adding 10 (*10-10*), 5 (*15-5*) or 1 (*19-1*) class in a single incremental step and performing two incremental steps adding 5 classes (*10-5*), five steps adding two

classes (*10-2*) and either ten (*10-1*) or five (*15-1*) steps adding one class. We follow the same approach as previous works and split the classes in alphabetical order.

Single-step incremental settings (10-10, 15-5, 19-1). Results are presented in Tab. 3.4, where we compare our method, MMA, with previous works that either use rehearsal [67, 52, 70] or not use it [188, 203, 145, 120]. We note that the former methods are not compared fairly with MMA, since we do not use any replay memory to store old samples. To ensure a fair comparison, we also report the results of ILOD [145] and Faster ILOD [120] using our same architecture and training protocol. Additionally, we provide two simple baselines: the joint training upper bound, where the architecture is trained using the whole dataset and all the annotations, and the fine-tuning, where the architecture is trained on the new data using Eq. (3.10), without employing any regularization strategy. We report the results in terms of mAP on old, new and all classes. The *Avg* metric equally weights the performance on both new and old classes by simply averaging their aggregated mAP.

As can be seen in Tab. 3.4, fine-tuning experiences a significant decrease in performance on the old classes, which demonstrates that catastrophic forgetting is a problem that needs to be addressed. Previous works have improved the performance by tackling the forgetting issue but MMA outperforms all of them, including those that use exemplars to prevent forgetting, thus validating our approach. When compared with ILOD [145] and Faster ILOD [120], our method achieves comparable performance on the old classes, but performs better on the new classes, leading to an improvement of 1% on both 19-1 and 15-5, and even 10% on the 10-10 setting. We believe that this improvement is mainly due to the revised distillation loss, which by modelling the missing annotations, eliminates inconsistent training objectives and thus increases the performance. When comparing MMA to other state-of-the-art methods, we observe that it outperforms the competitive rehearsal strategies in every setting without using exemplars. On the 19-1 setting, MMA outperforms ILOD-Meta by 0.5% when considering all classes equally (*1-20*) and by 1.1% OW-DETR when considering old and new classes equally (*Avg*). Similarly, in the 15-5 and 10-10 settings, MMA outperforms the best rehearsal method by 0.9% and 0.3% on all the classes, and by 0.7% and 0.3% on the *Avg* metric, respectively.

Multi-step incremental settings (10-5, 10-2, 15-1, 10-1). We consider a more realistic setting where we perform multiple incremental steps adding new classes to evaluate the ability of MMA to alleviate catastrophic forgetting. We compare the behavior of MMA against three baselines: fine-tuning, ILOD [145], and Faster ILOD [120], all implemented following our experimental protocol. The results for the four considered settings are reported in Tab. 3.5, showing the mAP% over multiple incremental steps and Fig. 3.6, where the

Table 3.6 Ablation study of the contribution of MMA components in the 15-5 setting. Results are mAP@0.5%. MMA is in green.

Eq. (3.12)	ℓ_{dist}^{RCN}	ℓ_{dist}^{RPN}	**1-15**	**16-20**	**1-20**	**Avg**
-	-	-	14.2	59.2	25.4	36.7
✓	-	-	40.0	57.8	44.4	48.9
✓	UKD	-	67.3	60.3	65.6	63.8
✓	l2	✓	**73.7**	56.8	69.5	65.3
✓	CE	✓	72.8	59.4	69.5	66.1
✓	UKD	✓	73.0	**60.5**	**69.9**	**66.7**

results after the last incremental step are displayed. Additionally, Tab. 3.5 reports the average performance across multiple steps *Avg-S*.

It can be observed that carrying out multiple incremental steps is difficult and existing methods exhibit a significant decrease in performance when compared to single step scenarios. Fine-tuning the network on new data without using any technique to prevent forgetting leads to the complete forgetting of old classes, resulting in performances close to 0% on old classes at the last step. ILOD [145] and Faster ILOD [120] can effectively reduce catastrophic forgetting, resulting in better results on both old and new classes. However, when compared to MMA, both ILOD and Faster ILOD achieve poorer results. After the last step, MMA is seen to obtain better performances on novel classes: +2.4% on 10-5, +3.3% on 10-2, +6.3% on 15-1, and +6.8% on 10-1 compared to the best among the baselines. Additionally, MMA also achieves comparable or greater performance than previous methods on the old classes. Overall, MMA outperforms the best among ILOD and Faster ILOD by 0.9% on 10-5, 2.7% on 10-2, 2.8% on 15-1, and 6.5% on the 10-1 setting. It is noteworthy that the improvement is greater when more classes are added, indicating that our method is better suited for multiple-incremental steps. From the trend over multiple training steps in Fig. 3.6, it can be seen that MMA is always comparable or better than previous methods. Notably, MMA largely outperforms the other methods when the number of training steps is large, as seen in the 10-1 setting.

Ablation Study. In Table 3.6, we present an in-depth analysis of our contributions, considering the 15-5 setting for incremental object detection. We evaluate each proposed component separately: the revised classification loss (Eq. (3.12)), the classification head knowledge distillation loss (ℓ_{dist}^{RCN}), the use of the RPN distillation loss (ℓ_{dist}^{RPN}), and finally, the use of a feature distillation loss, as proposed in [120]. The first row indicates the results of simply fine-tuning the network on the new data, without applying any regularization. It can be seen that the results are poor on the old classes, while it achieves good performance on the new

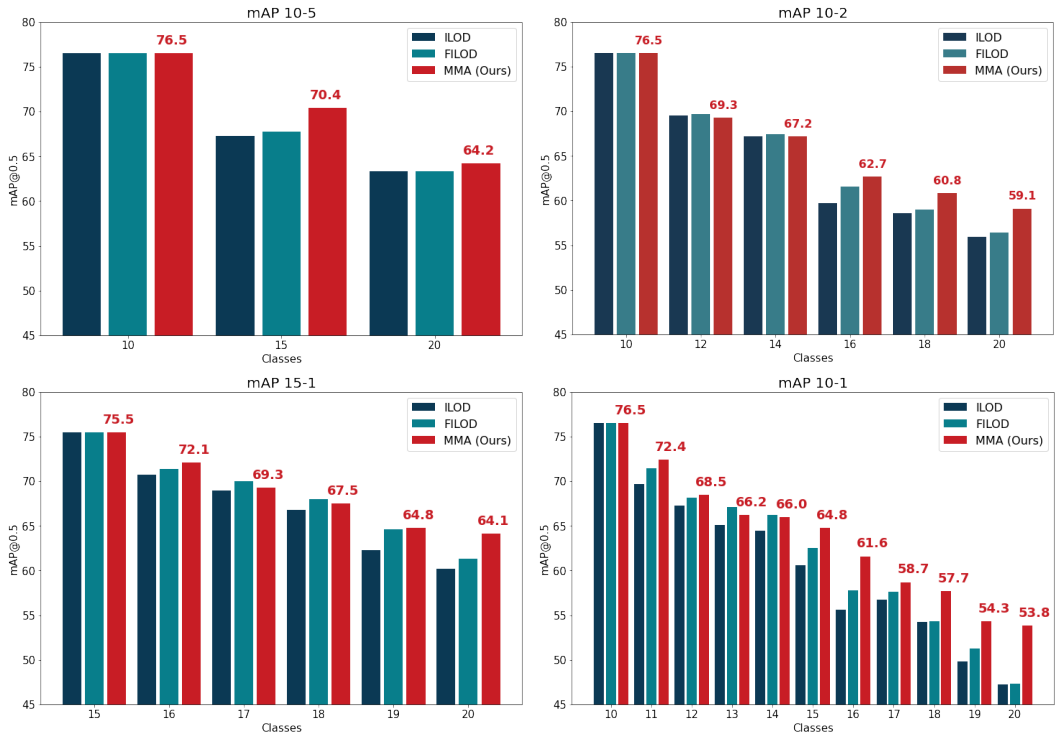


Fig. 3.6 mAP% results on multiple incremental steps on Pascal-VOC 2007.

ones. By adding the revised classification loss, the performance on the old classes substantially improves: from 14.2% to 40.0%. This is due to the handling of missing annotations that alleviates forgetting. By introducing the revised distillation loss presented in Eq. (3.15) (UKD), the performance is further increased, both on the old classes reaching 67.3% and on the new classes, rising from 57.8% to 60.3%. We believe that the improvement on the new classes is due to the distillation loss, as the model is able to better differentiate the old classes from the new ones, thus increasing the overall accuracy. The RPN distillation loss is then introduced, leading to the final MMA model. We observe that the performance on the old classes is further improved, reaching 73.0%, while the performance on the new classes remains comparable.

Finally, we compare the revised knowledge distillation in MMA with other possible choices. We draw inspiration from previous works and employ the L2 loss on the normalized classification scores [145, 120] and the cross-entropy (CE) loss between the probability of old classes [81]. We observe that MMA distillation performs better than these alternatives, particularly on the new classes, which clearly demonstrates the importance of modeling the missing annotations in order to learn them properly. On the average of old and new class

Table 3.7 mAP@(0.5,0.95)% results of incremental instance segmentation on Pascal-VOC 2012.

	19-1				15-5			
Method	**1-19**	**20**	**1-20**	**Avg**	**1-15**	**16-20**	**1-20**	**Avg**
Joint Training	40.4	54.1	41.1	47.2	41.0	41.2	41.1	41.1
Fine-tuning	6.7	46.3	8.7	26.5	1.9	35.3	10.2	18.6
Fine-tuning w/ Eq. (3.12)	12.5	47.5	14.3	30.0	13.0	35.5	18.6	24.2
ILOD [145]	40.1	38.3	40.0	39.2	39.2	30.8	37.1	35.0
Faster ILOD [120]	40.6	38.1	40.4	39.3	39.4	30.3	37.1	34.8
MMA	40.6	43.0	40.8	41.8	38.2	33.7	37.1	35.9
MMA + ℓ_{dist}^{MASK}	41.0	42.8	**41.1**	**41.9**	40.2	32.2	**38.2**	**36.2**

performance, MMA achieves 66.7%, 1.4% and 0.6% more than when using the L2 loss or the cross-entropy loss, respectively.

Instance Segmentation Results

We evaluate our method in instance segmentation considering two experimental settings: adding one (*19-1*) and five (*15-5*) classes in a single training step. As in object detection, we follow the alphabetical order of the dataset. Following the standard practice on instance segmentation, we report the mAP averaged across 11 IoU thresholds, ranging from 0.5 to 0.95, with a step of 0.05. We compare our method, MMA, with fine-tuning, fine-tuning using the revised classification loss (Eq. (3.12)), ILOD [145] and Faster ILOD [120]. For all the methods we employ the same architecture and hyperparameters.

Table 3.7 shows the results for the 19-1 and 15-5 settings, reporting the average mean Average Precision (mAP) of new and old classes separately, the average over all classes, and the average of new and old classes (*Avg*), with them weighted equally. We can observe that fine-tuning results in a significant amount of forgetting on old classes for both the 19-1 and 15-5 settings. Introducing the revised classification loss (Eq. (3.12)) helps to alleviate forgetting, but the results are still low for old classes, indicating that a technique to prevent forgetting is necessary. ILOD and FasterILOD do improve the performance on old classes, but at the cost of a decrease in performance on novel classes: they both lose nearly 8% on the 19-1 and 5% on the 15-5 with respect to fine-tuning. In contrast, our proposed MMA clearly improves the performance, preventing forgetting while also showing good performance on novel classes. In particular, compared to ILOD and Faster ILOD, MMA obtains, on new classes, nearly +5% and +3%, respectively on 19-1 and 15-5, while showing comparable performance on old classes. Considering the extended version of MMA (MMA + ℓ_{dist}^{MASK}), it slightly improves the performance on old classes compared to MMA, while obtaining comparable results on the new ones. Overall, it obtains 41.1% and 38.2% on the 19-1 and

15-5, respectively, 0.3% and 0.9% better than MMA. Interestingly, we note that, without any regularization on the mask head (MMA), we can still achieve good segmentation performance. This is due to the non competitiveness among classes on the mask head, which only regresses a binary segmentation mask, while the class is predicted by the classification head, as in standard Faster R-CNN. Overall, MMA and its extension demonstrate to outperform the other baselines in instance segmentation, showing a good balance between learning the new classes and avoiding forgetting the old ones.

3.4 Conclusion and Future Works

In this chapter, we investigated the catastrophic forgetting issue in image segmentation and object detection tasks. We identified that, due to the presence of multiple classes in every image and the fact that only annotations for novel classes are provided, the forgetting issue is exacerbated by the presence of non-annotated old and future classes in the background.

In the first part, we investigated the incremental class learning problem in the context of semantic segmentation, analyzing the semantic shift of the background class. To tackle this challenge, we presented the MiB method, that introduces a novel objective function and a classifier initialization strategy that enable the network to explicitly handle the semantic shift of the background, effectively learning new classes without compromising its performance in recognizing old ones. Our results indicate that our approach outperforms previous ICL methods by a significant margin, across both small and large datasets.

In the second part, we considered the issue of missing annotations for old and future classes in incremental learning for the object detection task, a problem that was overlooked by previous works. Missing annotations in object detection mislead the model to consider them as background region, effectively exacerbating forgetting on the old classes and making harder to learn classes that will appear in the future. We tackled this issue by extending the method designed in the previous section to object detection. The novel method, nicknamed MMA, revisits the standard distillation framework to consider non annotated regions as possibly containing past or future classes. Our approach outperforms the previous works on the Pascal-VOC 2007 dataset, considering muliple class-incremental settings. Futhermore, MMA exceeds methods employing rehearsal learning without using any sample from the past. Finally, we showed an extension of MMA in the instance segmentation task, achieving a new state of the art.

We hope that our work will set a new trend in the incremental learning community where we go beyond the simple image classification task and consider more realistic and challenging tasks. As a future work, we aim to study the property of state-of-the-art transformer architectures in these tasks, possibly extending the study to other challenging tasks, such as panoptic segmentation.

Chapter 4

Few-Shot or Zero-Label Semantic Segmentation

4.1 Introduction

The high cost of collecting and annotating images at the pixel level in order to train semantic segmentation models is a significant obstacle that restricts its applications. In the previous chapter, we examined the challenge of incremental learning in semantic segmentation (Sec. 3.2) under the assumption that a substantial number of labeled images were available for each training step. However, in reality, images depicting new classes are seldom available in large quantities, and it may not be financially feasible to annotate them due to budget constraints. In this chapter, we explore two different settings in semantic segmentation with the goal of expanding the knowledge of a trained segmentation model using the fewest possible images, or even just by providing a textual description of the new class.

In the first section, we introduce a new setting for incremental few-shot segmentation (iFSS), where a model is tasked with segmenting new classes over time with only a small number of images available and without forgetting the old ones. We show that previous approaches from similar research fields have not effectively addressed the challenges presented in this scenario: learning new classes without overfitting or forgetting previous knowledge. To address both of these issues, we have developed a framework called PIFS, which combines prototype-learning and knowledge distillation. We demonstrate the effectiveness of our approach on a novel benchmark comprising two datasets and multiple incremental scenarios.

In the second section, we take a step further and attempt to segment novel classes with only a textual descriptor provided for them. This setting, known as Zero-Label Semantic Segmentation, has been explored in previous studies [185, 15]. However, an important aspect has been overlooked: some classes may appear in the background of images in the dataset. To address this issue, we propose a new method called STRICT, which enhances existing state-of-the-art approaches by incorporating an iterative self-training technique with a consistency constraint to learn unannotated classes from available images.

The work presented in this chapter led to the publication of two works:

- Cermelli, F., Mancini, M., Xian, Y., Akata, Z., and Caputo, B. *Prototype-based incremental few-shot segmentation*. In Proceedings of the 2021 British Machine Vision Conference.
- Pastore, G., Cermelli, F., Xian, Y., Mancini, M., Akata, Z., and Caputo, B. *A closer look at self-training for zero-label semantic segmentation*. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (Workshop) (pp. 2693-2702).

4.2 Incremental Few-Shot Segmentation

Collecting images and providing pixel-level annotations for real-world applications is often a prohibitive cost. Ideally, we want to design segmentation models that are able to learn new classes over time requiring only a few annotated images. Inspired by object detection [122] and image classification [47] literature, in this section we aim to address the problem introducing the practical scenario of Incremental Few-Shot Segmentation (iFSS). iFSS, illustrated in Fig. 4.1,captures the challenges of previous settings and it has the goal to learn a segmentation model able to learn new classes with few samples (as in Few-Shot Semantic Segmentation (FSS) [143, 128, 37, 173, 195, 146]), while retaining good performance on previous knowledge (as in Generalized FSS methods (GFSS) [185]) and without access to old training data (as in Incremental Learning (IL) [103, 21]).

To evaluate iFSS, we have created an evaluation protocol and benchmark on two datasets, varying the number of classes, images per class, and learning steps. We observe that both IL and FSS methods are challenged when applied to this scenario, either focusing on not forgetting old knowledge [103, 21] or failing to adapt the representation to the new classes [47, 125, 146]. To address this issue, we propose **P**rototype-based **I**ncremental **F**ew-Shot **S**egmentation (PIFS), which is the first method to combine prototype learning [47, 125] with knowledge distillation [57]. PIFS exploits prototypes learning to incorporate new classes from a few images, aggregating the pixel-level features of new classes to impring them as weights on the classifier. Differently from existing few-shot methods [47, 125], during the few-shot learning (FSL) steps the network is fine-tuned end-to-end to extend the feature representation to account for the new classes. To prevent both overfitting and forgetting, we introduce a novel prototype-based distillation loss that integrates new class probabilities in the objective function. Additionally, we find that batch normalization [64] negatively affects the performance in iFSS when using only a few iages, since data are no more *i.i.d.*. We solve this issue replacing the standard normalization with batch-renormalization [63] in the FSL steps training. Experiments demonstrate that PIFS consistently outperforms the baselines.

Our contributions are as follows. (1) We introduce the iFSS problem, which focuses on learning from a limited number of images [143, 128, 37] while avoiding catastrophic forgetting [102, 81, 21]. (2) We propose PIFS, which surpasses the shortcomings of IL and FSL methods on iFSS by combining prototype learning (to initiate end-to-end training in the FSL steps), knowledge distillation (incorporating new class scores to reduce forgetting and prevent overfitting), and batch-renormalization (to address non-*i.i.d.* few-shot data). (3) We construct an extensive benchmark for iFSS and demonstrate that PIFS consistently

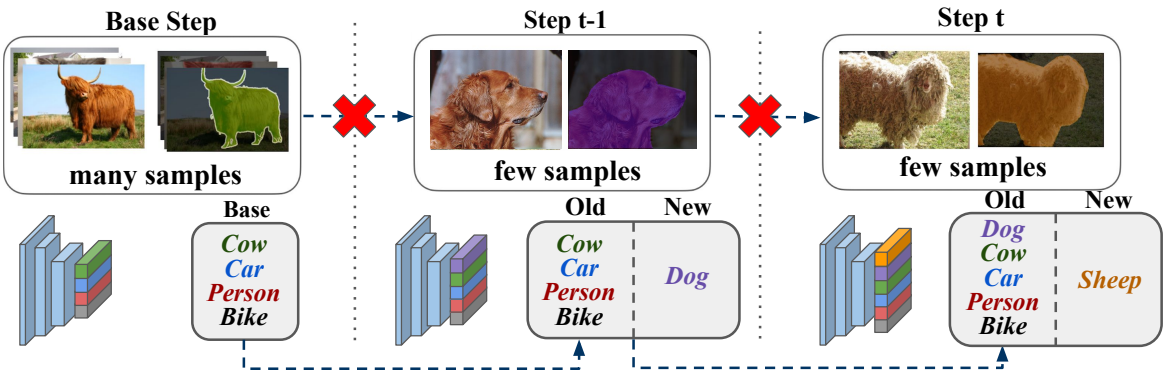


Fig. 4.1 Illustration of iFSS. A model is initially trained on a large labeled dataset to acquire a set of base classes. Subsequently, for few-shot learning, it is able to segment new classes with only a few annotated images and without access to the original datasets.

outperforms several IL and FSL methods on it. The code is available at http://github.com/fcdl94/FSS.

4.2.1 Related Works

Few-shot Learning is the task of training models able to classify a set of classes from a few samples. The two main approaches are optimization-based [130, 44, 110, 140] and metric-learning [147, 47, 168, 152, 27, 125]. Optimization-based methods are related to meta-learning, where a network is trained for a large variety of tasks such that it can solve new learning tasks using only a small number of training samples. Metric-learning approaches, on the other hand, have the goal of constraining the network embedding space such that instances of the same class are close to each other, leading to fast adaptation to new classes by already having a good representation. Recently, [27] demonstrated that fine-tuning a classifier based on a fixed feature extractor, trained on a large number of classes, achieve comparable performance to both optimization-based and metric-learning approaches while being much simpler. PIFS is related to the metric-learning approaches, in particular to [147, 47, 125], which learn to extract per-class prototypes from few images which are then used in the classification layer.

Few-shot Segmentation is the extension of few-shot learning for the semantic segmenation task. The standard approach for this task is metric-learning [143, 128, 37, 195, 173, 146, 197]. [37] adapted [147] to semantic segmentation by aggregating pixel-level feature representations to generate prototypes. [195, 173, 197] proposed refinement and iterative techniques to improve the prototypes exploiting the few available images. While these methods focuses

Table 4.1 Comparing different semantic segmentation settings. t denotes the current learning step, $\mathscr{C}^t$ denotes all classes labeled in the dataset $\mathscr{T}^t$ while $\mathscr{Y}^t = \cup_{s=0}^{t} \mathscr{C}^s$.

Semantic Segmentation	Training		Output	
	data	few-shot	class	multi-step
Offline	$\mathscr{T}^0$	-	$\mathscr{Y}^0$	-
Few-Shot [143, 128, 37, 195]	$\mathscr{T}^t$	✓	$\mathscr{C}^t$	-
Generalized Few-Shot [185]	$\cup_{s=0}^{t} \mathscr{T}^s$	✓	$\mathscr{Y}^t$	-
Incremental Learning [103, 21]	$\mathscr{T}^t$	-	$\mathscr{Y}^t$	✓
Incremental Few-Shot	$\mathscr{T}^t$	✓	$\mathscr{Y}^t$	✓

only on learning novel classes, [146] proposed to update also the old classes prototypes while computing the ones for new classes. Inspired by these works, PIFS employs prototype-learning to generate an initialization for new-class classifier weights but, differently, it fine-tunes the whole network using a distillation loss to reduce overfitting and forgetting.

Benchmarks. There are multiple settings with the goal of learning new classes in the segmentation task, which shares some similarities with the proposed iFSS scenario. In Few-Shot Segmentation (FSS) [143, 128, 37, 195, 173, 146, 197] the aim is to segment new classes given only a few images depicting them. However, in FSS the training is performed using an episodic scenario [168], focusing on segmenting only a novel class depicted by annoteted images in the support set, often leading to a binary [143, 128, 146, 195] or 2-way [37, 173, 197] segmentation problem, which is not practical for real use-cases. [185] proposed Generalized Few-Shot Segmentation (GFSS) to overcome this issue. The goal of GFSS is to segment train a network to classify both new and previously seen classes, learning them respectively from a few and several annotated images. [185] focuses on an offline setting, assuming that all the images can be accessed at every training step, that is often unfeasible (Sec. 2.2). In contrast to these settings, we have seen in Sec. 3.2 that incremental learning in semantic segmentation assumes to have a large dataset for training the new classes without access to old datasets. The proposed iFSS setting relies on the intersection of these benchmarks, requiring to learn new classes from a small dataset without accessing old data. We note that there are settings similar to iFSS proposed for image classification [47, 156, 134], object detection [122], and instance segmentation [46] but no previous work extended the task to semantic segmentation. We summarize the differences between iFSS and previous existing settings in Tab. 4.1.

4.2.2 Incremental Few-Shot Segmentation (iFSS)

The objective of iFSS is to learn a model that assigns each pixel in an image its corresponding semantic label in the set $\mathscr{Y}$. This set is expanded over time with only a few pixel-level annotated images for the new classes. Formally, let us denote $\mathscr{Y}^t$ as the set of semantic categories known by the model after learning step t, where *learning step* denotes a single update of the model's output space. A sequence of datasets $\{\mathscr{T}^0,\dots,\mathscr{T}^T\}$ is received during training, where $\mathscr{T}^t = \{(x,y)|x \in \mathscr{X}, y \in (\mathscr{Y}^t)^N\}$. We recall that x represents an image in the space $\mathscr{X} \in \mathbb{R}^{N\times 3}$, with N is the number of pixels in the image ($N = |\mathscr{I}|$). Each training step introduces a set of novel classes $\mathscr{C}^t$ such that $\mathscr{C}^i \cap \mathscr{C}^j = \emptyset$ for $i \neq j$ and $\mathscr{Y}^t = \bigcup_{s=0}^{t} \mathscr{C}^s$. The first dataset $\mathscr{T}^0$ is large and contains multiple images for a many classes while all other datasets $\mathscr{T}^t$ are few-shot ones; that is, $|\mathscr{T}^0| \gg |\mathscr{T}^t|$, for all $t \geq 1$. The model begins by being trained on the large dataset $\mathscr{T}^0$, then incrementally updated with few-shot datasets. We refer to this first learning step on $\mathscr{T}^0$ as the "base" step. It should be noted that at step t, only dataset $\mathscr{D}^t$ is available to the model. Two assumptions are made from this formulation: i) each dataset provides annotations only for new classes; ii) pixels of old classes in $\mathscr{Y}^{t-1}$ are labeled in dataset $\mathscr{T}^t$, but *only* if present. These assumption are different from the previous work Sec. 3.2 since only a few images need to be fully annotated, being feasible also on small budget.

4.2.3 Prototype-based iFSS

In this section, we introduce **P**rototype-based **I**ncremental **F**ew-Shot **S**egmentation (PIFS), which is illustrated in Fig. 4.2. During the base step, PIFS learns a prototype-based model using a standard training procedure. For the few-shot learning (FSL) steps, it first uses prototype learning to initialize the weights of the classifiers for new classes and then fine-tunes the network end-to-end with a prototype-based distillation loss while utilizing batch-renorm to handle non-*i.i.d.* data.

Learning a prototype-based model. Our purpose is to discover a model f_{θ^t} that associates each pixel with a probability distribution across the set of classes, *i.e.* $f_{\theta^t} : \mathscr{X} \to \mathbb{R}^{|I|\times|\mathscr{Y}^t|}$, where t stands for the last learning step. We assume $f_{\theta^t} = g^t \circ e^t$ is composed of a feature extractor $e^t : \mathscr{X} \to \mathbb{R}^{N\times d}$ and a classifier $g^t : \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times|\mathscr{Y}^t|}$, where d is the feature dimension and g^t is a softmax classifier with parameters $W^t = [w_1^t,\dots,w_{|\mathscr{C}^t|}^t] \in \mathbb{R}^{d\times|\mathscr{Y}^t|}$. This definition encompasses most state-of-the-art segmentation architectures such as [26, 200]. At the initial step, we want to prepare f_{θ^0} to include new classes utilizing few examples. To do so, we force the classifier weights to represent class prototypes. The prototypes should

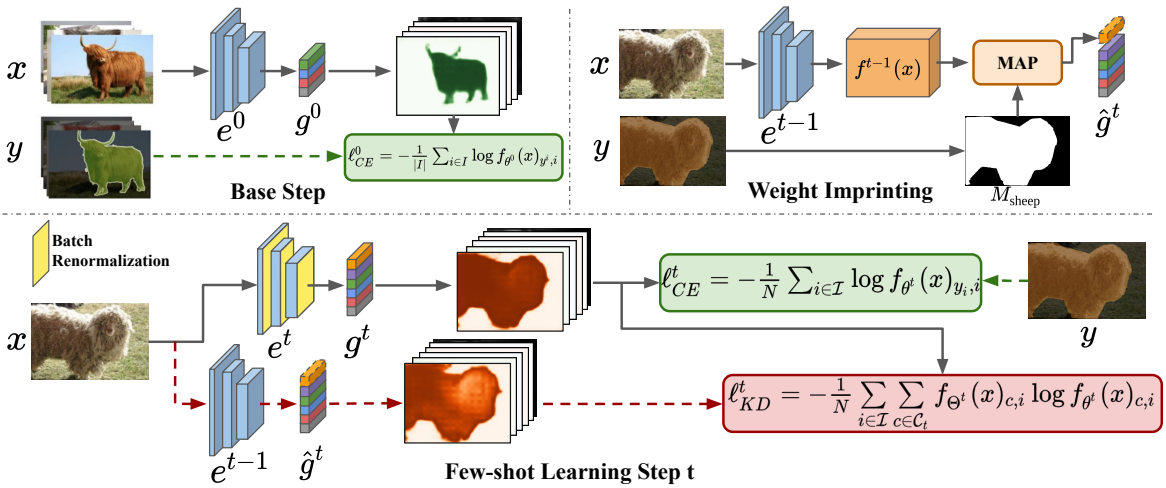


Fig. 4.2 Illustration of PIFS. Initially, in the base step (top left) we train a prototype-based model with the cross-entropy loss l_{CE}. When few images of a new class are available (top-right), we use Masked Average Pooling (MAP) to initialize the prototypes. We then fine-tune the network (bottom) with both the cross-entropy loss and our prototype-based knowledge-distillation (l_{KD}). To tackle the non-*i.i.d.* few-shot data,we employ batch-renorm in the few-shot learning steps.

reflect the average pixel-level features of a class, so that the features extracted from (few) pixels of the new classes give an accurate estimation of their respective classifier weights. In accordance with prior work [47, 125], we achieve this through a cosine classifier. In the base learning step, we employ the cross-entropy loss over all pixels to train the network:

$$\ell^0_{CE}(x,y) = -\frac{1}{N}\sum_{i\in\mathscr{I}} \log f_{\theta^0}(x)_{y_i,i} \tag{4.1}$$

where $f_{\theta^0}(x)_{c,i}$ indicates the class c probability for the i-th pixel of x. To compute $f_{\theta^t}(x)_{c,i}$ we use a normalize using the softmax the cosine similarity between the features and the class prototype w^t_c:

$$f_{\theta^t}(x)_{c,i} = g^t(e^t(x))_{c,i} = \frac{\exp(s^t_{c,i})}{\sum_{k\in\mathscr{C}^t}\exp(s^t_{k,i})}, \quad s^t_{c,i} = \tau\frac{e^t_i(x)^\intercal w^t_c}{||e^t_i(x)||\;||w^t_c||} \tag{4.2}$$

where $e^t_i(x)$ denotes the features obtained for the pixel i, and τ is a temperature value that scales the cosine similarity in the range $[-\tau,\tau]$. The loss in Eq. (4.1) forces the model to minimize the cosine distance between a class prototype and its feature representation, ensuring their compatibility for the few-shot learning steps.

Initializing prototypes of new classes. At the few-shot learning step, our goal is estimate the prototype w_k for the class $k \in \mathscr{C}^t$ employing the features representation of it from the

dataset $\mathscr{T}^t$. Thus, the new class prototypes are computed aggregating the features extracted for each pixel of the class k present in images of $\mathscr{T}^t$. To this aim, we follow previous works [146, 37, 173] employing masked average pooling (MAP) to initialize the prototypes:

$$w_k^t = \text{MAP}_k(\mathscr{T}^t) = \frac{1}{|\mathscr{T}_k^t|} \sum_{(x,y)\in\mathscr{T}_k^t} \frac{\sum_{i\in\mathscr{I}} M_{k,i}(y) \frac{e_i^t(x)}{||e_i^t(x)||}}{\sum_{i\in\mathscr{I}} M_{k,i}(y)}, \tag{4.3}$$

where $M_k(y)$ is a binary mask that indicates which pixels belong to class k, and $\mathscr{T}_k^t$ is a subset of the dataset $\mathscr{T}^t$ where each image contains at least one pixel of class k. Despite being simple, this strategy demonstrated to provide a good estimate of the class representation that can be used as a classifier. Note that it is not needed in standard IL, where multiple images are availabe, but is crucial in iFSS, where starting from random weights would lead to overfitting the few images.

Prototype-based Distillation. Despite the feature extractor is able to extract good prototypes to initialize the classifier, it may have a sub-optimal representation of the novel classes, given their difference with the previous ones. To improve its expressivity, in the few-shot learning steps we fine-tune the model *end-to-end*. However, as demonstrated in previous works [47, 122], end-to-end training using only a few images may lead to overfitting the new classes and forgetting the old ones. To address both issues, we design a novel distillation loss that avoid forgetting by regularizing the model output while keeping into account the prototypes for the new classes to avoid overfitting.

We train the model in the few-shot learning steps using the following objective function. Given a pair $(x,y) \in \mathscr{T}^t$, we compute:

$$\ell^t(x,y) = \ell_{CE}^t(x,y) + \lambda \ell_{KD}^t(x, f_{\theta^t}, f_\Theta) \tag{4.4}$$

where λ is a hyperparameter, ℓ_{CE}^t represents the loss in Eq. (4.1) over $\mathscr{Y}^t$. ℓ_{KD}^t employs f_Θ as a teacher model to compute the knowledge distillation loss. In previous works [103, 21] the teacher was a copy of the model frozen after the previous learning step, that is $f_\Theta = f_{\theta^{t-1}}$. Differently, we aim to fully exploit the properties of prototype learning and we define f_Θ as the model after the inizialization of the prototypes ofr the new classes. Formally, we set $f_\Theta = \hat{f}_{\theta^t}$, where $\hat{f}_{\theta^t} = \hat{g}^t \circ e^{t-1}$ and the parameters $\hat{W}^t = [\hat{w}_1^t, \dots, \hat{w}_{|\mathscr{C}^t|}^t]$ of $\hat{g}^t$ as:

$$\hat{w}_k^t = \begin{cases} w_k^{t-1}, & \text{if } k \in \mathscr{C}^{t-1} \\ \text{MAP}_k(\mathscr{T}^t) & \text{otherwise.} \end{cases} \tag{4.5}$$

The distillation loss ℓ^t_{KD} is then defined as:

$$\ell^t_{KD}(x, f_{\theta^t}, f_\Theta) = -\frac{1}{N}\sum_{i\in,\mathscr{I}}\sum_{c\in\mathscr{C}^t} f_\Theta(x)_{c,i}\log f_{\theta^t}(x)_{c,i}. \tag{4.6}$$

Note that in Eq. (4.6), the teacher produces scores for both old classes in $\mathscr{Y}^{t-1}$ and *new* ones in $\mathscr{C}^t$ that are used in the distillation loss. This presents two advantages w.r.t. standard knowledge distillation in IL: (i) it alleviates forgetting by forcing the current model to keep scores for old classes similar to the old model, (ii) it encourages to maintain the prototypes of new classes close to their initial value given by e^{t-1}, reducing overfitting on the few images of new classes.

Coping with non-i.i.d. data. Despite the regularized training, we found that in extreme few-shot scenarios (*e.g.* 1-shot settings) a main cause of the drop in performance is the fact that the data does not follow anymore the independent and identically distributed (*i.i.d.*) assumption: we have small datasets where most pixels belong to new classes, thus the input is not identically distributed. However, having *i.i.d.* data, in fact, is crucial for standard normalization techniques, such as batch-normalization (BN) [64], that are frequently employed in neural networks. In practice, BN layers normalize each value of the input z^c_i, where c indicates the channel and i its spatial position, to have a learnable mean and standard deviation:

$$\hat{z}^c_i = \gamma\frac{z^c_i - \mu^c}{\sigma^c} + \beta, \tag{4.7}$$

where γ and β are the learnable parameters, and μ^c and σ^c are, respectively, the mean and standard deviation across all the images in the batch on the channel c. During inference, the BN layers replace the batch statistics with the running statistics, which are the moving averages of the mean μ^c_r and standard deviation σ^c_r:

$$\hat{z}^c_i = \gamma\frac{z^c_i - \mu^c_r}{\sigma^c_r} + \beta. \tag{4.8}$$

A non *i.i.d.* input leads to a drift in the statistics of the BN layers in the network, making them poor and biased and harming the performance.

Two simple solutions to solve this issue are either freezing and using the global BN statistics of the base step both at training and inference, or using the one computed on the new dataset during training but without their value at infrerence time. We found that the former solution causes important training instability and the latter misaligns the features extracted for the new classes at training and test time, leading to poor performance. Ideally,

during the FSL steps the features should be normalized without: i) shifting the statistics toward the new class data, harming old class performance; ii) disaligning the training and inference normalization statistics. To solve this issue, taking inspiration from the incremental learning literature [95], we propose the use of batch-renorm (BR) [63]. BR revisits BN by normalizing a feature c with the running statistics in place of the batch statistics:

$$\hat{z}_i^c = \gamma\left(\frac{z_i^c - \mu^c}{\sigma^c}\frac{\sigma^c}{\sigma_r^c} + \frac{\mu^c - \mu_r^c}{\sigma_r^c}\right) + \beta, \tag{4.9}$$

where μ_r^c and μ^c are the global and batch mean, and σ_r^c and σ^c the global and batch standard deviation. It is important to note that $\frac{\sigma^c}{\sigma_r^c}$ and $\frac{\mu^c - \mu_r^c}{\sigma_r^c}$ are treated as constants for the purposes of gradient computation . Moreover, after the base step we freeze μ_r^c and σ_r^c to avoid a shifting toward the new class statistics that would damage the model performance.

4.2.4 Experiments

Experimental Protocol. In order to assess the performance of a model, we need a large dataset containing an initial set of classes and one or more few-shot datasets containing new classes. We create such an experimental setting on the Pascal-VOC 2012 dataset containing 20 classes, and the COCO dataset containing 80 thing classes. Following previous work on FSS [195, 173], we consider 15 and 60 of the classes as *Base* and 5 and 20 as *New*, for the VOC and COCO datasets respectively. We propose two protocols, each starting with pretraining on the *Base* classes: in one there is a single FSL step on all *New* classes, while in the other we have multiple steps: 5 steps of 1 class on the VOC dataset and 4 steps of 5 classes on the COCO dataset. We divide the VOC dataset into 4 folds of 5 classes and the COCO dataset into 4 folds of 20 classes, running experiments 4 times by considering each fold in turn as the set of new classes. We name the single-step settings VOC-SS and COCO-SS, and the multi-step VOC-MS and COCO-MS.

We looked at how well the model performs on different settings, using 1, 2 or 5 images in the FSL step. We averaged the results of multiple trials, each using a different set of images. The images were randomly sampled from the set of images containing at least one pixel of the new class, without imposing any constraint about the presence of old classes. We only used the provided few-shot images (both for weight-imprinting and for training) without using other images. To ensure that the model does not use pixels from new classes in the base step, we excluded from the initial dataset all the images containing pixels of new classes. Finally, we report the results on the whole validation set of each dataset, considering all the

seen classes. We assessed a methods performance using three metrics based on the mean intersection-over-union (mIoU) as in GFSS [185]: mIoU on base classes (*mIoU-B*), mIoU on new classes (*mIoU-N*), and the harmonic mean of the two (*HM*). As in [21, 103], we report all the results after the last training step.

Baselines. We consider nine baselines to cmpare our method: three few-shot classification (FSC) methods [125, 47, 161], two (G)FSS methods [146, 185], three IL methods [81, 103, 21] and naïve fine-tuning (FT). The models we compare are either state-of-the-art in the setting they were proposed [47, 161, 146, 185, 103, 21] or simple yet effective baselines (e.g. [125, 81]). FSC methods that we employ are Weight-imprinting [125] (WI) and Dynamic WI [47] (DWI), that inject prototypes in the classification layer of the network; Rethinking FSL [161] (RT), that emply a frozen representation and only fine-tunes the classifier for new classes. From the FSS literature, we adapt Adaptive Masked Proxies [146] (AMP), a WI variant updating also classification weights of the old classes, and Semantic Projection Network [185] (SPN), a method designed for GFSS that projects the feature representation in a semantic space , such as word embeddings. Finally, the IL methods are Learning without Forgetting (LwF) [81], applying a standard distillation (KD) [57] on the class probabilities; Incremental Learning Techniques (ILT) [103], that performs KD at feature-level; and Modeling the Background (MiB) that has been described in Sec. 3.2. Note that, for MiB, the revised cross-entropy of MiB reduces to the standard cross-entropy formulation when old classes are annotated.

Implementation details. We use the Deeplab-v3 [28] with ResNet-101 [55] for all the experiments. To reduce the memory footprint, we follow the implementation of [138], unifying normalization and activation functions. We consider as feature extractor the ResNet-101 followed by the ASPP and as classifier the 1×1 final convolutional layer. The ResNet-101 has been initialized on ImageNet dataset following the standatd practice of FSS and ILSS [37, 143, 173, 185, 146, 21]. We train all the methods in the base step with a learning rate 10^{-2} and batch size 24 for 30 epochs on Pascal-VOC and 20 epochs on COCO. In settings with a single FSL step, we update the model using batch size $min(10, |\mathscr{T}_n|)$ for 1000 iteration with learning rate 10^{-3}. In settings with multiple FSL steps, we fine-tune the model using a batch size equals to $min(10, |\mathscr{T}_n|)$ for 200 iteration each step, employing a learning rate of 10^{-4}. To ensure fair comparison, all the baselines have been re-implemented by us using the same segmentation network and training hyperparameters. The results are reported using single-scale full-resolution images, without applying any post-processing step.

iFSS: Single few-shot learning step. Table 4.2 reports the results on the single few-shot learning (FSL) step setting of 5 classes on VOC-SS and of 20 classes on COCO-SS. PIFS

Table 4.2 iFSS: mIoU on single few-shot learning step scenarios.

		VOC-SS									COCO-SS								
		1-shot			2-shot			5-shot			1-shot			2-shot			5-shot		
	Method	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM
	FT	58.3	9.7	16.7	59.1	19.7	29.5	55.8	29.6	38.7	41.2	4.1	7.5	41.5	7.3	12.4	41.6	12.3	19.0
FSC	WI [125]	62.7	15.5	24.8	63.3	19.2	29.5	63.3	21.7	32.3	43.8	6.9	11.9	44.2	7.9	13.5	43.6	8.7	14.6
FSC	DWI [47]	**64.3**	15.4	24.8	**64.8**	19.8	30.4	64.9	23.5	34.5	44.5	7.5	12.8	45.0	9.4	15.6	44.9	12.1	19.1
FSC	RT [161]	59.1	12.1	20.1	60.9	21.6	31.9	60.4	27.5	37.8	**46.2**	5.8	10.2	**46.7**	8.8	14.8	46.9	13.7	21.2
FSS	AMP [146]	57.5	16.7	25.8	54.4	18.8	27.9	51.9	18.9	27.7	37.5	7.4	12.4	35.7	8.8	14.2	34.6	11.0	16.7
FSS	SPN [185]	59.8	16.3	25.6	60.8	26.3	36.7	58.4	**33.4**	42.5	43.5	6.7	11.7	43.7	10.2	16.5	43.7	15.6	22.9
IL	LwF [81]	61.5	10.7	18.2	63.6	18.9	29.2	59.7	30.9	40.8	43.9	3.8	7.0	44.3	7.1	12.3	44.6	12.9	20.1
IL	ILT [103]	**64.3**	13.6	22.5	64.2	23.1	34.0	61.4	32.0	42.1	**46.2**	4.4	8.0	46.3	6.5	11.5	**47.0**	11.0	17.8
IL	MiB [21]	61.0	5.2	9.7	63.5	12.7	21.1	**65.0**	28.1	39.3	43.8	3.5	6.5	44.4	6.0	10.6	44.7	11.9	18.8
	PIFS	60.9	**18.6**	**28.4**	60.5	**26.4**	**36.8**	60.0	**33.4**	**42.8**	40.8	**8.2**	**13.7**	40.9	**11.1**	**17.5**	42.8	**15.7**	**23.0**

Table 4.3 iFSS: average mIoU across steps on multi few-shot learning step scenarios.

		VOC-MS									COCO-MS								
		1-shot			2-shot			5-shot			1-shot			2-shot			5-shot		
	Method	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM
	FT	47.2	3.9	7.2	53.5	4.4	8.1	58.7	7.7	13.6	38.5	4.8	8.5	40.3	6.8	11.6	39.5	11.5	17.8
FSC	WI [125]	66.6	16.1	25.9	66.6	19.8	30.5	66.6	21.9	33.0	**46.3**	8.3	14.1	**46.5**	9.3	15.5	46.3	10.3	16.9
FSC	DWI [47]	**67.2**	16.3	26.2	**67.5**	21.6	32.7	**67.6**	25.4	36.9	46.2	9.2	15.3	**46.5**	11.4	18.3	**46.6**	14.5	22.1
FSC	RT [161]	49.2	5.8	10.4	36.0	4.9	8.6	45.1	10.0	16.4	38.4	5.2	9.2	43.8	10.1	16.4	44.1	16.0	23.5
FSS	AMP [146]	58.6	14.5	23.2	58.4	16.3	25.5	57.1	17.2	26.4	36.6	7.9	13.0	36.0	9.2	14.7	33.2	11.0	16.5
FSS	SPN [185]	49.8	8.1	13.9	56.4	10.4	17.6	61.6	16.3	25.8	40.3	8.7	14.3	41.7	12.5	19.2	41.4	18.2	**25.3**
IL	LwF [81]	42.1	3.3	6.2	51.6	3.9	7.3	59.8	7.5	13.4	41.0	4.1	7.5	42.7	6.5	11.3	42.3	12.6	19.4
IL	ILT [103]	43.7	3.3	6.1	52.2	4.4	8.1	59.0	7.9	13.9	43.7	6.2	10.9	47.1	10.0	16.5	45.3	15.3	22.9
IL	MiB [21]	43.9	2.6	4.9	51.9	2.1	4.0	60.9	5.8	10.5	40.4	3.1	5.8	42.7	5.2	9.3	43.8	11.5	18.2
	PIFS	64.1	**16.9**	**26.7**	65.2	**23.7**	**34.8**	64.5	**27.5**	**38.6**	40.4	**10.4**	**16.5**	40.1	**13.1**	**19.8**	41.1	**18.3**	**25.3**

achieves the top results on every dataset and shot, outperforming the best IL method by 3.2% and 5.6% in HM, and the best FSL one by 6% and 2.6%, on VOC-SS and COCO-SS respectively. SPN [185] is comparable with PIFS on 2 and 5 shot settings (+0.1 HM on VOC-SS 2-shot), but it uses external informations (word embeddings) to improve generalization on new classes. However, PIFS outperforms SPN with a margin of +2.8% HM on VOC-SS, and +2% HM on COCO-SS for 1-shot settings, demotrating its superiority in this challenging setting. Some methods (e.g., DWI, ILT) surpass PIFS in terms of mIoU-B metric; however, they achieve suboptimal results when it comes to new classes due to either fixed representations (e.g., DWI) or lack of prototype learning (e.g., ILT). While suffering a slight decrease in mIoU-B, PIFS shows the best performance for new classes while providing an optimal tradeoff between learning and remembering capabilities. Figure 4.3 compares qualitative results on VOC-SS 1-shot for different methods; WI and DWI with fixed representations either focus too much on context (e.g., *horse*, third row) or assign pixels to related classes (e.g., *bicycle* vs *motorbike*, second row), which is also observed in ILT and SPN (e.g., *dog*, last row). On the other hand, PIFS provides precise segmentation masks even when train samples differ significantly from test ones (e.g., *cat*, last row).

iFSS: Multiple few-shot learning steps. Table 4.3 reports the average performance obtained under multiple FSL step settings, that is 5 steps of 1 class (VOC-MS) and 4 steps of 5 classes (COCO-MS). VOC-MS is an extremely challenging setting: it contains few training images

Fig. 4.3 Qualitative results on the VOC-SS 1-shot setting.

(as little as one in the 1-shot case) that belongs only to one class, resulting in a clearly non-*i.i.d.* setting. The PIFS approach has demonstrated a marked improvement compared to the baseline methods, achieving an average 12.9% and 5.0% improvement in HM over the best IL method on VOC-MS and COCO-MS respectively, and outperforming the best FSS method by 8.3% and 0.9% on the same datasets. Additionally, PIFS has consistently outperformed all non end-to-end methods, including WI, DWI, AMP, and RT, by an average of 2% on new classes, showcasing the effectiveness of fine-tuning in even the most challenging scenario with only one training image. On the other hand, FT failed in this scenario, emphasizing the importance of prototype learning and the distillation loss in avoiding overfitting on new classes and preserving knowledge of old classes. IL methods have struggled in learning new classes, showing improvement only in COCO-MS 2 and 5 shots due to their knowledge distillation losses (for example, a 1.2% improvement in HM on 2-shot and 5.1% improvement on 5-shot for ILT). However, they still lag behind PIFS, with a 7% and 2.4% reduction in HM on COCO-MS 2-shot and 5-shot respectively. The gap between PIFS and the best IL method becomes even more pronounced in VOC-MS and 1-shot settings, with PIFS outperforming the best IL method by 20.5% in HM on VOC-MS and 26.7% in HM on COCO-MS. This is due to the difficulty of learning new classes from scratch with limited training images

Table 4.4 Ablation of the different component of PIFS. WI: weight imprinting. BR: batch-renormalization. PD: prototype-based distillation loss. KD: [81]. L2: [103].

				VOC-SS 1-shot			COCO-SS 1-shot		
FT	WI	BR	ℓ_{KD}	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM
✓				58.3	9.7	16.7	41.2	4.1	7.5
✓			KD	61.5	10.7	18.2	**43.9**	3.8	7.0
✓			L2	61.3	10.4	17.8	43.3	3.3	6.1
	✓			**62.7**	15.5	24.8	43.8	6.9	11.9
✓	✓			56.6	14.0	22.5	39.9	7.4	12.5
✓	✓		PD	57.6	14.7	23.4	40.5	7.9	13.2
✓	✓	✓		59.9	17.7	27.4	39.8	7.4	12.5
✓	✓	✓	KD	62.1	18.2	28.1	41.6	7.4	12.6
✓	✓	✓	L2	61.9	18.4	28.3	41.2	7.0	12.0
✓	✓	✓	PD	60.9	**18.6**	**28.4**	40.8	**8.2**	**13.7**

without leveraging prototype learning. Finally, SPN has shown a tendency to forget when learning from small datasets (e.g., VOC-MS 1-shot), while PIFS still outperformed it (i.e. +12.8% HM) despite not utilizing external knowledge.

Ablation study. In this section, we report an ablation study assessing the contribution of the method components. We compared prototype initialization (WI) with a standard random classifier, end-to-end training (FT), batch-renorm (BR) instead of batch normalization, and our prototype knowledge distillation (PD) to standard ones, such as KD, distilling on old class probabilities [81], and L2, acting on the features extracted from e^{t-1} and e^t, losses. The results on the challenging 1-shot benchmarks of VOC-SS and COCO-SS can be seen in Tab. 4.4. The results of FT, FT+KD, and FT+L2 indicate that starting with random weights in the classifier leads to poor performance on new classes. However, WI alone achieves good results by leveraging prototype learning and avoiding forgetting. When the initialized network is trained (FT+WI), there is an improvement compared to FT alone (at least +5% in HM), but there is also a decrease in performance on base classes (nearly 6% and 4% HM on COCO-SS and VOC-SS, respectively) due to forgetting. The table shows that both PD and BR can mitigate forgetting. PD improves results on base classes in both datasets, while BR is particularly effective when few images are available (27.4% HM on VOC-SS). Furthermore, when applied together, they lead to the best performance on both datasets (13.7% HM on COCO-SS). Finally, we compared our distillation loss (PD) to KD and L2 losses. Although combining them with WI improves performance, our PD loss outperforms both (1.7% HM over L2 and 1.1% HM over KD on COCO-SS), demonstrating the importance of designing a distillation loss that also reduces overfitting of new class prototypes.

Table 4.5 iFSS: mIoU on single few-shot learning step scenarios with background shift. PIFS* uses the revised cross-entropy loss defined in Sec. 3.2.

		VOC-SS-strict									**COCO-SS-strict**								
		1-shot			**2-shot**			**5-shot**			**1-shot**			**2-shot**			**5-shot**		
	Method	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM	mIoU-B	mIoU-N	HM
	FT	55.0	10.2	17.2	55.5	19.2	28.6	43.7	26.8	33.2	35.3	4.5	8.0	32.8	7.4	12.1	26.9	11.1	15.7
FSC	WI [125]	62.7	15.5	24.8	63.3	19.2	29.5	63.3	21.7	32.3	43.8	6.9	11.9	44.2	7.9	13.5	43.6	8.7	14.6
FSC	DWI [47]	**64.3**	15.4	24.8	**64.8**	19.8	30.4	64.9	23.5	34.5	44.5	7.5	12.8	45.0	9.4	15.6	44.9	12.1	19.1
FSC	RT [161]	60.1	11.0	18.6	62.3	19.7	29.9	61.0	26.0	36.5	**46.0**	4.0	7.3	**46.5**	5.1	9.2	**46.8**	7.5	13.0
FSS	AMP [146]	56.6	16.6	25.7	54.6	18.8	28.0	51.6	18.2	26.9	42.7	6.8	11.8	42.7	8.2	13.7	42.4	10.0	16.2
FSS	SPN [185]	56.4	16.4	25.4	57.1	25.3	35.1	48.7	30.2	37.3	38.1	7.0	11.8	37.0	10.4	16.3	33.2	15.1	20.8
IL	LwF [81]	60.6	11.2	18.9	62.8	19.5	29.8	56.2	29.7	38.9	43.0	4.5	8.1	42.6	8.3	13.9	40.6	13.7	20.5
IL	ILT [103]	63.1	14.1	23.0	63.6	23.8	34.7	58.9	31.6	41.2	45.2	5.1	9.2	45.0	8.0	13.6	44.0	13.3	20.4
IL	MiB [21]	61.0	6.1	11.1	63.6	13.7	22.6	**65.0**	29.4	40.5	43.7	4.2	7.7	44.2	7.1	12.3	44.4	13.8	21.1
	PIFS	59.1	**18.3**	**27.9**	58.8	26.2	36.2	57.2	32.6	41.5	34.9	**8.9**	14.2	34.6	11.7	17.4	32.6	15.6	21.1
	PIFS*	60.3	18.0	27.8	60.3	**26.3**	**36.6**	59.6	**33.1**	**42.5**	38.8	8.8	**14.4**	39.2	**11.8**	**18.1**	38.4	**16.1**	**22.6**

iFSS with annotations only for new classes. Despite annotating at pixel-level only a few images highly reduces the labeling cost, it might be not feasible in some scenarios, such as when annotation requires an expert (such as a doctor in medical images).In this section, we assess the ability of our model to learn new classes without forgetting when old class pixels are not annotated in the FSL steps. Note that this setting follows the definition of ILSS in Sec. 3.2, where old class pixels are considered as background, introducing the issue of the shifting semantic of the background class, exacerbating catastrophic forgetting. We follow the *disjoint* protocol described in Sec. 3.2, where we exclude all the the images containing pixels of new classes from the base step.

To better deal with this setting, we introduce PIFS*, that replace the cross-entropy loss with the revised cross-entropy loss introduced in Sec. 3.2, effectively addressing the background-shift issue. The results for the single-step configurations of VOC (VOC-SS-strict) and COCO (COCO-SS-strict) are shown in Tab. 4.5. These results consider 1, 2 or 5 images in the FSL steps. The results show that PIFS and PIFS* attain the best balance between learning and forgetting, obtaining the highest HM across all configurations. In particular, PIFS* outperforms the best IL method by an average of 2.7% and 4% in terms of HM for VOC and COCO, respectively, and outperforms the best FSL method by 5.7% and 2.5% for the same datasets. SPN struggles to handle the background shift, resulting in poor performance in mIoU-B, where PIFS* outperforms it by 6% on VOC and 2.7% on COCO on average. Methods that only calculate classifier weights from new class pixels (i.e. WI, DWI) remain unaffected by old class annotations and have the same performance as the non-strict setting (Tab. 4.2). However, PIFS outperforms the best of them (DWI) by 5.7% and 2.5% on average in HM for VOC and COCO, respectively. Comparing PIFS and PIFS*, it is observed that addressing the background shift helps in avoiding forgetting the old classes, as demonstrated by the higher mIoU-B results obtained by PIFS*: it outperforms PIFS by 1.7% on VOC and 4.7% on COCO on average.

4.3 Zero-label Semantic Segmentation

A challenging goal of deep learning is to generate models able to predict novel classes during test time without requring any additional training step. This field of study is known in literature as zero-shot learning or zero-label semantic segmentation (ZLSS) [185, 15, 51] when applied to the semantic segmentation task. The ZLSS setting assumes that a model is trained on a dataset containing a set of *seen* classes for which pixel-level annotations are provided. Then, during evaluation, the model is required to predict *unseen* classes, *i.e.* classes for which it has never being provided any label during training. Despite the challenging task, the ZLSS goal is to segment only unseen classes, discarding and forgetting all the accumulated knowledge of seen classes. For this reason, to truly learn novel classes and increase the model capabilities over time, the Generalized Zero-Label Semantic Segmentation (GZLSS) setting has been proposed with the goal of evaluating segmentation models on both seen and unseen classes. This setting is extremely challenging since it introduce a sever class-imbalance issue: the model has seen multiple examples of seen classes during training but it has been never explicitly trained to predict the unseen classes, leading to a significant performance drop on them. Previous methods address the issue either by using external information for the unseen classes [185] or by employing a generative model that synthesize features of unseen classes [51, 15]. Despite they obtained promising results, previous works did not considered that the training set may include many unlabeled pixels from unseen classes due to the large amount of class co-occurrences in semantic segmentation, that can be exploited to improve performance on unseen classes.

We propose to capture the latent information about unseen classes by supervising the model with self-produced pseudo-labels for the unlabeled pixels. Generating accurate pseudo-labels for unseen classes is extremely challenging since the model is highly uncertain on them since it has never received supervision on these classes. Consequently, using a pretrained GZLSS model (e.g., SPNet [185]) produce very noisy pseudo-labels and may compromise the performance (as shown in Figure 4.4). To reduce the noise and produce better pseudo-labels, we introduce an efficient **S**elf-**Tra**i**n**ing with Consistency **C**onstrai**n**t (STRICT) method: we consider a pseudo-label as correct only if the model predict it on multiple augmented versions of the image and we periodically update the pseudo-label generator, progressively improving the performance on the unseen classes. Through an extensive experimental study, we show that STRICT outperforms previous works on two different dataset, PascalVOC12 and COCO-stuff, achieving a new state of the art.

Our main contributions are: (a) we devise STRICT, a method that employs a self-training pipeline to obtain strong supervision for unseen classes from unlabelled pixels in GZLSS

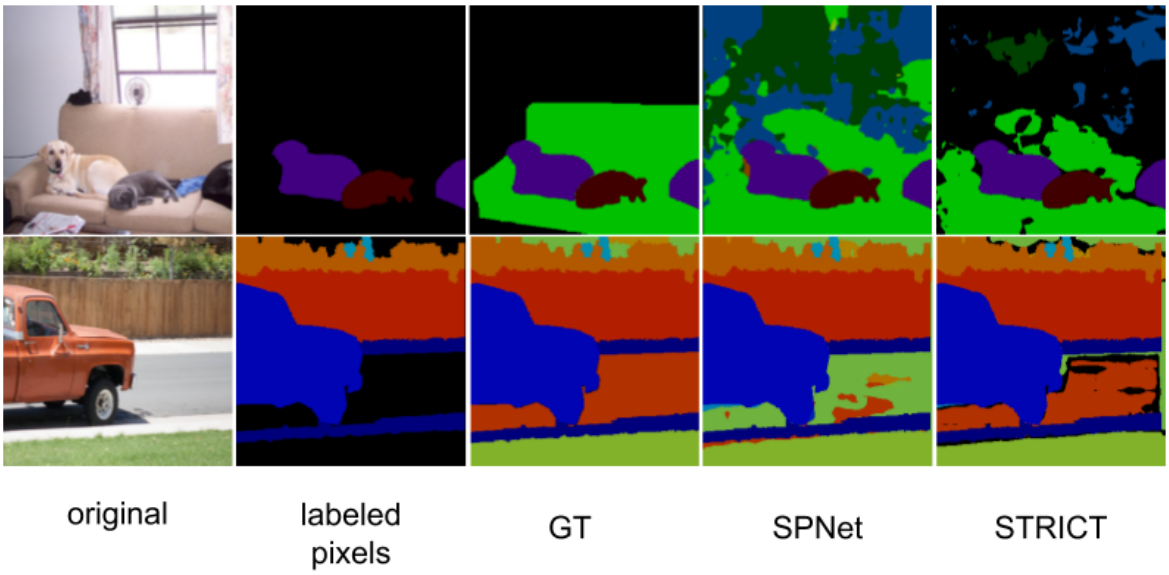


Fig. 4.4 In generalized zero-label semantic segmentation, pixels not annotated are ignored although they might be relevant at test-time since they belong to unseen classes. We propose to pseudo-label the unlabeled pixels on training images employing the Self-Training with Consistency Constraint (STRICT) method. *Labeled pixels* and *GT* refers to the masked and actual ground truth, respectively. *SPNet* and *STRICT* indicates the pseudo-labeled masks produced by SPNet [185] and STRICT.

exploiting a consistency constraints on different image augmentations; (b) we show that a model fine-tuned through such process progressively enhances its ability in predicting unseen classes, and consequently the quality of pseudo-labels; (c) we extensively analyze our approach on two datasets, outperforming previous works.

4.3.1 Related Works

Zero-Shot Learning. The models for Zero-Shot Learning (ZSL) can be grouped into four categories based on their approach for transferring knowledge from seen to unseen categories [183]. The first category utilizes a two-stage method to calculate posterior class probabilities using intermediate attributes obtained from images, which are further processed by additional classifiers [74]. The second category views the task as a problem of visual-semantic embedding, evaluating the compatibility between the visual and semantic spaces, such that proximity indicates a semantic relationship [182, 5, 4, 116, 199]. The third category employs a class-level semantic conditioned generator to provide synthetic CNN features for unseen classes during the training of a discriminative classifier [184, 14]. The final category addresses the task in a purely generative manner, modeling the class-conditional distributions to capture semantic relationships between seen and unseen classes [10, 166, 79, 108].

Generalized Zero Label Semantic Segmentation. Only three methods directly tackle GZLSS: SPNet [185], ZS3 [15], and CaGNet [51]. SPNet adopts an approach of the second category. Inspired by [116], it uses a segmentation model to extract visual features, which are then projected into semantic features through matrix multiplication with a word embedding representation. ZS3 and CaGNet extend the feature-generative method used in [14] for classification. ZS3 uses a Graph Convolutional Network to incorporate contextual prior knowledge about category relationships (e.g. "mouse is near the keyboard"), while CaGNet does the same but at a pixel level, feeding the feature generator with a contextual latent code. While ZS3 and CaGNet do not directly address GZLSS, they offer an extension of their approach that do so through self-training (ZS5 and CaGNet + ST, respectively). In this study, we build upon SPNet since it is a simple and flexible approach and we demonstrate how to improve its prediction capabilities on unseen classes.

Self-training in semantic segmentation. Pseudo-labeling has been widely used as a self-supervision strategy in poorly annotated computer vision scenarios [75, 9, 123, 151, 65]. The concept of self-training based on consistency has been widely used in the field of semi-supervised semantic segmentation [205, 106, 113, 29]. In the PseudoSeg method [205], pseudo-labels are generated for unlabeled pixels by fusing different predictions from the decoder and Grad-CAM. The consistency of these predictions is then imposed on multiple augmented images. On the other hand, the approach presented in [106] involves adversarial training of a segmentation model that serves as a generator to improve the predictions for unlabeled data. The discriminator is used to distinguish between real and fake predictions and also as a measure of quality to select the most confident predictions. In [113], the predictions are made invariant over different encoder's output perturbations. Lastly, [29] demonstrates that iteratively applying pseudo-labeling improves scene segmentation in urban video sequences. In addition, self-training has been employed in transductive ZSL and GZLSS. [196] operates in a transductive ZSL scenario, generating pseudo-labels using the model confidence. In transductive GZLSS, [87, 9] employ an unbiased loss to filter and enhance the quality of pseudo-labels. On the other hand, ZS5 and CaGNet filter out a percentage of the less confident pseudo-labels produced for unlabeled pixels. Similarly, we aim to produce pseudo-labels for unseen classes on unlabeled pixels improving the robustness and generalization ability of GZLSS methods.

4.3.2 STRICT: Self-training with Consistency Constraints

Problem definition. Let $\mathscr{S} = \{1,\dots,\mathscr{C}^s\}$ and $\mathscr{U} = \{\mathscr{C}^s+1,\dots,\mathscr{C}^s+\mathscr{C}^u\}$ denote respectively the disjoint label spaces of seen and unseen classes. $\mathscr{T} = \{(x,y)|x \in \mathscr{X}, y \in \{\{\mathtt{b}\} \cup \mathscr{S}^N\}$ is the training set where x is an image made of N pixels and y is its corresponding label mask whose value y_i at each pixel $i \in \mathscr{I}$ ($N = |\mathscr{I}|$) is its corresponding class label belonging to one of the seen classes $\mathscr{S}$ or the unlabeled class denoted as $\mathtt{b}$. For each class is also provided a word embedding (e.g., word2vec [105]) associated to its class name. We denote the word embedding matrices of seen and unseen classes with $W^s \in \mathtt{R}^{D\times|\mathscr{C}^s|}$ and $W^u \in \mathtt{R}^{D\times|\mathscr{C}^u|}$, where D is the dimension of the word embedding space. Given $\mathscr{T}$, W^s and W^u, GZLSS has the goal to learn a network capable of making pixel-wise predictions among both seen and unseen classes.

Semantic projection network. SPNet [185] approach is made of a visual-semantic embedding network and a semantic projection layer. The former, denoted as e_θ, is a segmentation backbone (such as DeepLab [26]) that maps the input image x to the visual embedding space, that is $e_\theta(x) \in \mathtt{R}^{D\times N}$. The semantic projection layer computes the product between the visual embedding and word embeddings. Then, the softmax normalizes the outputs to obtain the posterior probability over the seen classes:

$$P(\hat{y}_i = c|x;W^s) = \frac{\exp(w_c^T e_\theta(x)_i)}{\sum_{c'\in\mathscr{S}}\exp(w_{c'}^T e_\theta(x)_i)} \tag{4.10}$$

where i is a pixel in $\mathscr{I}$, $w_c \in \mathtt{R}^D$ represents the c-th row of the matrix W^s and corresponds to the class c word embedding. The standard cross-entropy loss is computed for a training sample (x,y) as:

$$\ell_{CE} = -\frac{1}{N}\sum_{i\in\mathscr{I}} \mathbb{1}[y_i \neq \mathtt{b}]\log P(\hat{y}_i = y_i|x) \tag{4.11}$$

where y_i is the ground-truth at pixel i and $\mathbb{1}[y_i \neq \mathtt{b}]$ is the indicator function (1 if $y_i \neq \mathtt{b}$ otherwise 0). We remark that pixels from unseen classes might be present, but not labeled (i.e., $y_i = \mathtt{b}$), in the image x.

At test time, we obtain the model probabilities employing word-embeddings for both seen and unseen classes. The prediction is then obtained with the following:

$$\arg\max_{c\in\mathscr{S}\cup\mathscr{U}} P(\hat{y}_i = c|x;[W^s,W^u]) \tag{4.12}$$

Algorithm 1: STRICT algorithm.

$P_0 \leftarrow$ ZLSS pretrained model;
$P_t \leftarrow$ ZLSS model at iteration t;
$P_{t-1} \leftarrow$ ZLSS model at previous iteration (t-1);
$\{A_1(\cdot),...,A_k(\cdot)\} \leftarrow$ data augmentations;
$\mathscr{T} \leftarrow$ train set;
$T \leftarrow$ number of iterations;
for $t = 1,2,...,T$ **do**
  foreach (x,y) *in* $\mathscr{T}$ **do**
    $\hat{y} \leftarrow$ model prediction $P_{t-1}(x)$;
    $A \leftarrow$ augmentations $\{A_1(x),...,A_k(x)\}$;
    $\Gamma \leftarrow$ hard pseudo labeled masks$\{\bar{y}^k,...\bar{y}^K\}$;
    $\bar{y} \leftarrow A_1^{-1}(\bar{y}^k) \cap \ldots \cap A_K^{-1}(\bar{y}^K)$;
    $\ell \leftarrow \ell_{CE}(x,y) + \lambda \ell_{CE}(x,\bar{y})$;
    $P_t \leftarrow$ SGD model update;
  end foreach
  $P_{t-1} \leftarrow P_t$
end for

Consistent pseudo-label generation. SPNet [185] is trained optimizing Eq. (4.11) on the seen classes being given the training set $\mathscr{T}$ and word embeddings W^s. A major difference with standard ZSL setting, however, is that in the training set $\mathscr{T}$ appear labeled pixels of seen classes but also unlabeled pixels of unseen classes, which are not exploited by SPNet and ignored when computing the training loss. In practice, the unseen classes are actually present in the dataset but they do not contribute to the training. To take into account the presence of unseen classes, we introduce an effective pseudo-labeling technique to provide a pseudo-annotation for the unlabeled pixels in the training set, *i.e.* for pixels i in $\mathscr{T}$ with $y_i = \mathtt{b}$. We assume that the seen classes have been completely annotated in the training set, and the unlabeled pixels can only belong to unseen classes. The pseudo-labeling strategy thus produces labels only for the unseen classes. Specifically, for each of the training images containing unlabeled pixels, the pseudo-label generator G computes $\bar{y} = G(x)$, where $\bar{y} \in \{\{\mathtt{b}\}, \mathscr{U}\}^N$ denotes the pseudo-label mask for the image x and $\bar{y}_i = \mathtt{b}$ if the pixel i belongs to a seen class.

Directly obtaining the pseudo-labels from the model using its predictions is harmful since the model is highly noisy (as shown in Fig. 4.4). Inspired by previous works in the segmentation literature [113, 157, 129, 107, 73], we propose to reduce the noise by employing an approach based on consistency regularization. We exploit the simple principle that, if the model makes coherent prediction on multiple augmented versions of the same

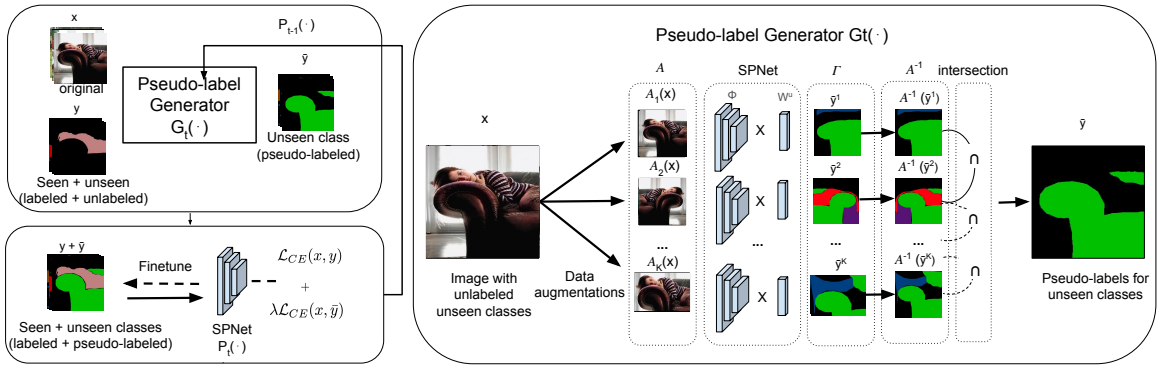


Fig. 4.5 An overview of STRICT: during the t-th iteration, the generator G_t produces a mask $\bar{y}^k$ for the unlabelled pixels of each of the K augmentations $\{A_1(x),\ldots,A_k(x)\}$. The final pseudo-label mask $\bar{y}$ is obtained computing as the intersection among them. The model P_t is fine-tuned with the pixel-wise cross-entropy loss computed both on labeled (y) and pseudo-labeled ($\bar{y}$) pixels. At the iteration $(t+1)$, P_t will be used for the pseudo-label generator.

image, it is very likely that the prediction is correct. Formally, providing the model with an image x containing unlabeled pixels, K different data augmentations (denoted as $A_k(\cdot)$) are applied to obtain K augmented images, denoted as $\{A_1(x),\ldots,A_K(x)\}$. A_1 indicates the identity mapping and we consider simple data augmentation techniques to generate the K versions such as horizontal mirroring and resize with different scaling factors. Intuitively, these data augmentations only transform the image spatially and the semantic content of each pixel remains the same. To generate the pseudo-labels, we compute the model predictions for every unlabeled pixels in each augmented image:

$$\bar{y}_i^k = \underset{c\in\mathscr{U}}{\arg\max}\, P(\hat{y}_i = c|A_k(x);W^u) \qquad \forall k\in\{1,\ldots,K\},\forall i\in\mathscr{I}. \tag{4.13}$$

This operation led to a set of K predictions $\{\bar{y}^k,\ldots,\bar{y}^K\}$. We then merge the K predictions to obtain the pseudo-label by applying the intersection operation:

$$\bar{y} = A_1^{-1}(\bar{y}^k)\cap\ldots\cap A_K^{-1}(\bar{y}^K), \tag{4.14}$$

where A_k^{-1} refers the inverse data augmentation function that reverts the distortions to return in the original pixel coordinates. The intersection operator filters out the prediction that are inconsistent across multiple image versions, reducing the noise in pseudo-labels. While similar consistency regularization techniques have been explored in semi-supervised literature [12], we are the first to apply the consistency constraints for the GZLSS task.

Iterative self-training. Our goal is to exploit the ground-truth and pseudo-labels to iteratively fine-tune the model to train it both on seen and unseen classes. Formally, we start from $t = 0$ with the model that is trained using only the ground-truth labels using Eq. (4.11). Then, in each following step, we first employ the pseudo-label generator to obtain, for each training image, a pseudo-label mask $\bar{y} \in \{\{\texttt{b}\}, \mathscr{U}\}^N$ of unseen classes that is complementary to the real label mask $y \in \{\{\texttt{b}\}, \mathscr{S}\}^N$ of seen classes. Using y and $\bar{y}$ we can in turn fine-tune the model, improving its performance on both seen and unseen classes. Formally, at each training step t we minimize the following loss:

$$\ell = \ell_{CE}(x, y) + \lambda \ell_{CE}(x, \bar{y}) \tag{4.15}$$

where ℓ_{CE} is the cross-entropy loss defined in Eq. (4.11) and λ is a hyperparameter. The first loss term is the SPNet loss that exploits the ground-truth annotations, while the second term utilizes the generated pseudo-labels on the unlabeled pixels. The loss provide supervision for the unseen class without requiring any annotated images for them and providing a balanced training set that improve performance in GZLSS. Algorithm 1 describes our iterative training pipeline in details.

To summarize, STRICT initialize the model using the SPNet method and is then composed of two iterative steps, as illustrated in Fig. 4.5: (1) twe generate pseudo-labels for unlabeled pixels and (2) feed the pseudo-labels back to the training set and retrain the model using Eq. (4.15). The two steps are repeated, meaning that the fine-tuned model will be used to generate more accurate pseudo-labels for retraining the model.

4.3.3 Experiments

Datasets and metrics. Our method is tested on two datasets, PascalVOC [43] and COCO-stuff [18], with data splits and validation procedure based on previous works [185, 51]. The train/val/test sets are composed of mutually exclusive classes: 12/3/5 classes on PascalVOC and 155/12/15 classes on COCO-stuff. To fine-tune, we use a two-stage procedure where we first select the best hyperparameters on the seen train classes and unseen validation classes, then train on both seen train and validation classes with fixed hyperparameters. For PascalVOC, we conduct experiments with the background as one of the seen classes. GZLSS performance is measured, following [185], in terms of mean Intersection over Union (mIoU) on seen ($\mathscr{S}$) and unseen ($\mathscr{U}$) classes, as well as the harmonic mean (HM) between them.

Method	PascalVOC			COCO-stuff		
	$\mathscr{S}$	$\mathscr{U}$	HM	$\mathscr{S}$	$\mathscr{U}$	HM
SPNet [185]	73.3	15.0	21.8	20.5	14.3	16.8
ZS3 [15]	77.3	17.7	28.7	34.7	9.5	15.0
CaGNet [51]	78.4	25.6	39.7	35.5	12.2	18.2
SPNet+ST [185]	77.8	25.8	38.8	34.6	26.9	30.3
ZS5 [15]	78.0	21.2	33.3	34.9	10.6	16.2
CaGNet + ST [51]	78.6	30.3	43.7	**35.6**	13.4	19.5
STRICT	**82.7**	**35.6**	**49.8**	35.3	**30.3**	**32.6**

Table 4.6 Comparing with the state of the art on PascalVOC and COCO-stuff.

Baselines and implementation details. We evaluated our approach against three GZLSS methods: SPNet [185], and two generative methods, ZS3 [15] and CaGNet [51]. We also included the self-training variants of CaGNet (CaGNet+ST) and ZS3 (ZS5), where the top percentage of pixels assigned to unseen classes were used as pseudo-labels. We additionally reported the results of another baseline, calibrated SPNet, trained through hard pseudo-labelling of unlabeled pixels without consistency strategy (SPNet+ST). To ensure fairness in comparison with previous works, we employed DeepLabV2 [27] as the segmentation model with ResNet-101 [55] as backbone, pretrained on Imagenet following in previous works [185]. We utilized SGD, with a momentum of 0.9 and a weight decay of $5 \cdot 10^{-4}$, and a polynomial decay with an initial learning rate of $2.5 \cdot 10^{-4}$, as in [27]. We initialized the network on seen classes training for 20K iterations on VOC and 100K iterations on COCO with a batch size of 8 images. We then fine-tuned the network with our self-training strategy, training one cycle of self-training for 2K iterations on PascalVOC12 and 22K iterations for COCO-stuff. The SPNet and SPNet+ST have been reimplemented.

Comparison with the state of the art

In Tab. 4.6, we present the results of comparing our approach with the state of the art on PascalVOC and COCO-stuff datasets. The experiments show that self-training strategies outperform the performance of all methods and for all metrics. For instance, on PascalVOC, ZS5 and CaGNet+ST improve their not self-trained counterparts by almost 5% and 4% respectively, while SPNet+ST improves the base SPNet by 17% in HM. Similarly, on COCO-stuff, ZS5 and CaGNet improve their performance by 1.5% and 1.3% respectively, while SPNet+ST surpasses all more complex generative approaches on unseen mIoU and HM. These results confirm that self-training is highly beneficial, especially for non-generative methods. Furthermore, our STRICT strategy outperforms all published results by a good

Table 4.7 PascalVOC results with background class included among the seen set.

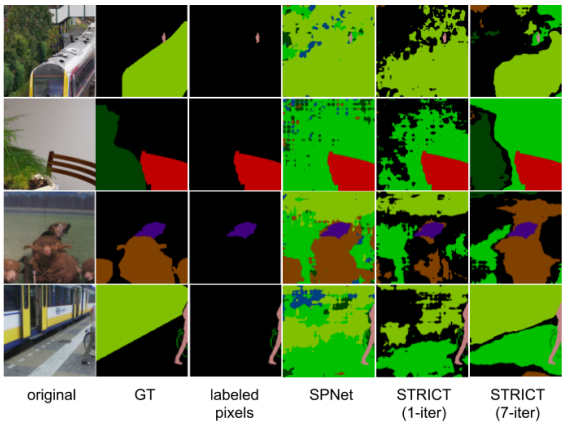
Method	PascalVOC		
	$\mathscr{S}$	$\mathscr{U}$	HM
SPNet [185]	54.7	2.5	4.7
ZS3 [15]	59.0	4.0	7.5
SPNet+ST [185]	72.7	4.0	7.6
ZS5 [15]	66.1	1.7	3.7
STRICT	**74.7**	**14.3**	**24.0**

Table 4.8 Ablation of different transformations for the consistency constraint of STRICT on PascalVOC.

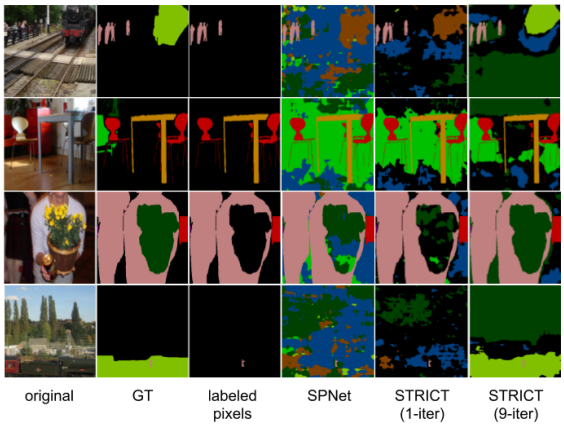
Mirroring	Scaling	$\mathscr{S}$	$\mathscr{U}$	HM
		77.8	25.8	38.8
✓		80.4	27.2	40.7
	down	82.1	27.8	41.5
	up	82.0	31.1	45.1
	random	81.6	29.4	43.2
✓	down	**83.7**	29.2	43.3
✓	up	82.5	**32.9**	**47.0**
✓	random	83.2	31.4	45.6

margin. On PascalVOC, it surpasses the previous state of the art (CaGNet+ST) by 6.1% on HM and 5.3% on unseen mIoU, and on COCO-stuff, it surpasses CaGNet by 16.9% on unseen class mIoU and almost 13.1% on HM. When compared with the SPNet+ST baseline, STRICT shows a higher improvement on PascalVOC and a less marked improvement on COCO-stuff. These improvements are significant and emphasize the importance of employing an effective self-training strategy for zero-label semantic segmentation models. Moreover, the self-training approach reduces the bias of the network on seen classes while not requiring a calibration term or generating pixel features, as in the case of SPNet and generative approaches, but rather exploiting the information coming from the unlabeled pixels.

Impact of the background on PascalVOC. The conventional GZLSS techniques used for se,antic segmentation do not take into account the differentiation between foreground and background, and only evaluate pixels of the foreground objects. We conducted an assessment of the impact on performance when the background category is incorporated in the class space. This scenario is notably more challenging as the pixels of unseen classes could be mislabeled as background, resulting in the model's inability to discriminate them effectively. Table 4.7 presents the results for our method, SPNet, ZS3, and their self-trained variations. All methods, including STRICT and the baselines, experience a significant decline in performance when



(a) Pseudo-labels generated with STRICT for PascalVOC unseen classes when background is ignored.



(b) Pseudo-labels generated with STRICT for PascalVOC unseen classes when background is included.

Fig. 4.6 Qualitative pseudo-labeling results of STRICT on PascalVOC without (left) and with (right) background as seen class. Train GT refers to labels for the unseen classes.

the background is included in the classifier. For instance, when comparing Tab. 4.6 to Tab. 4.7, we observe that SPNet achieves only 2.5% of mIoU on unseen classes, which is almost 12% lower than Tab. 4.6, with an overall 4.7% on the harmonic mean (17% lower). The results slightly improve with self-training, with SPNet+ST achieving 4% mIoU on unseen classes and a 7.6% of harmonic mean. ZS3 surprisingly outperforms its self-trained counterpart ZS5 in this setting since generating a robust classifier for unseen classes in a generative manner is challenging in segmentation because of the images' high complexity. The bias of the network towards predicting the background in place of unseen class pixels also hampers the generation and pseudo-labeling process. However, our STRICT approach is effective even in this scenario, with an mIoU on unseen classes of 14.3% and an overall harmonic mean of 24%. These results are comparable to the calibrated SPNet+ST performance in the standard scenario where the background is ignored, with the performance being only slightly lower (3% on harmonic mean and unseen class mIoU) than ZS3. Despite these promising outcomes, the performance gap between our model in the two scenarios remains significant (25% on harmonic mean and 21% on unseen mIoU), signifying that additional technical components are required to address the technical challenges of GZLSS for object segmentation when the background is included, explicitly addressing issues such as the semantic shift of the background class (see Sec. 3.2).

Qualitative results. In order to compare our model with other baselines, we present the qualitative semantic segmentation results of both our method and ZS5 in Fig. 4.8. The figure demonstrates that our model can accurately detect pixels of both seen (e.g. person) and

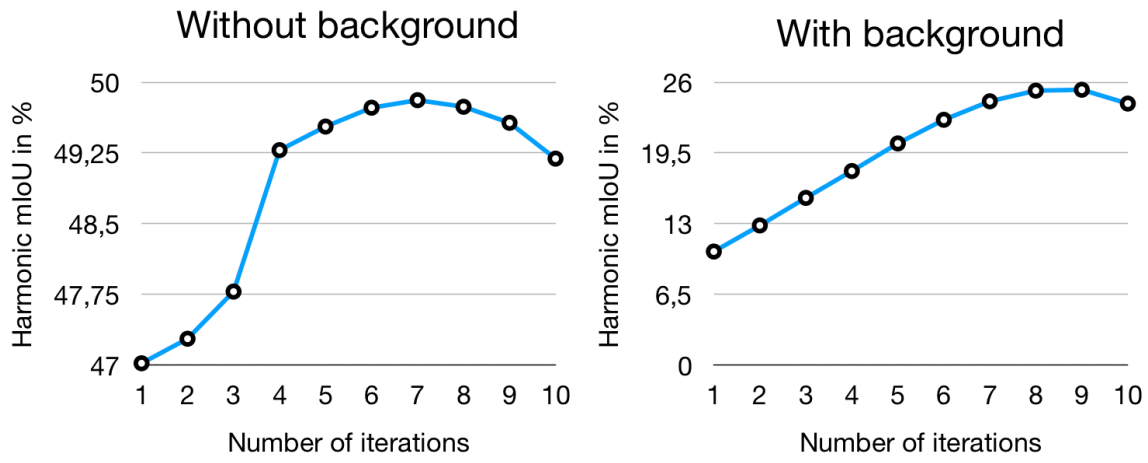


Fig. 4.7 STRICT mIoU along the iterative self-training procedure.

unseen (e.g. sofa) categories, and achieves a good balance between the two. For example, in the second row of the left image, ZS5 misidentifies most of the pixels of the unseen class *sheep* as a seen class *cow*, revealing its inclination towards seen classes. In contrast, our model segments the *sheep* almost perfectly, with only a few misclassified pixels. Similarly, ZS5 erroneously classifies the unseen class *table* as a seen class *tv*, while our model almost correctly segments it. These images also highlight a limitation of our approach, namely, the results depend on the number of co-occurring pixels. For instance, since the *plant* class occupy a small area of the images, it is challenging for the network to generate consistent pseudo-labels, resulting in lower recognition ability for that class. Future research may consider strategies to regularize the supervision for unseen classes based on the number of pseudo-labels generated for each of them.

Ablation study

Different image transformations. We perform a study regarding the most effective image transformations for implementing the self-training with consistency constraints. We specifically examine straightforward and reversible image-level changes, such as three versions of multi-scaling (reducing, enlarging, and random scaling) and mirroring. The findings of our evaluation are presented in Tab. 4.8. Overall, multi-scaling generally proves to be more advantageous than only employing mirroring. Among the scaling options, increasing the image size yields the best outcomes, with the highest mIoU on unfamiliar categories (31.1%) and HM (45.1%). By integrating mirroring with upscaling, we achieve the most exceptional performance, with a 32.9% mIoU on unfamiliar classes and a 47% HM.

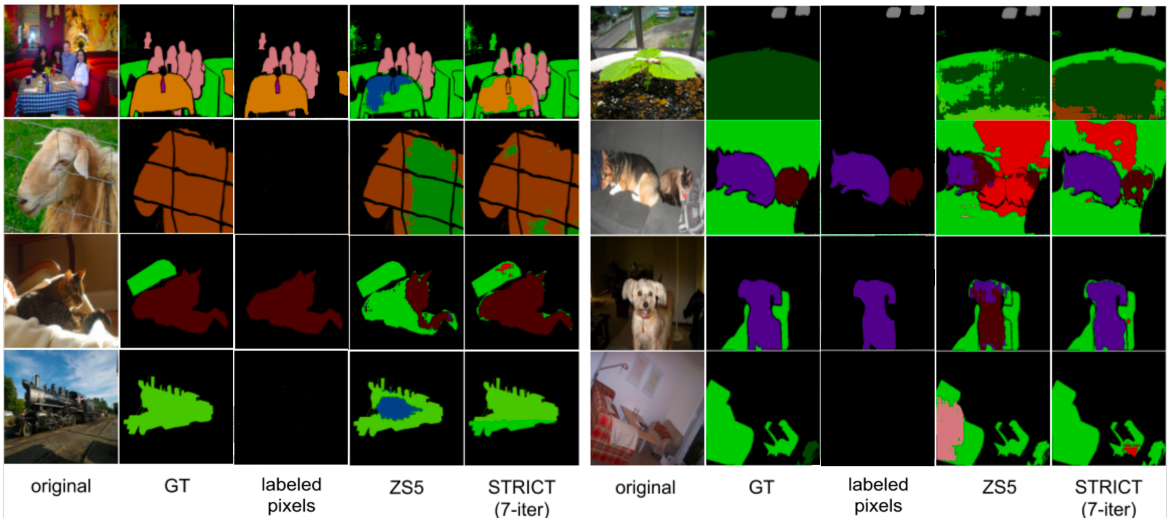


Fig. 4.8 Qualitative comparison of STRICT on PascalVOC.

Number of self-training iterations. The self-training procedure, which involves updating the pseudo-labeling model after each iteration, is a crucial element of our algorithm. In Fig. 4.7, we examine the impact of the number of self-training iterations on STRICT. We present the results as mIoU on unseen classes and harmonic mean, with and without background. The findings reveal that the performance of both metrics and settings tends to improve with an increase in self-training iterations. Notably, performance gains are rapid until six iterations, following which they plateau or slightly decline. This decline may be due to the absence of ground-truth for unseen class pixels, resulting in noisy predictions that are partially but not entirely eliminated by our consistency constraint.

Qualitative analysis on pseudo-labels. Generating good pseudo-labels for our model on unseen class pixels is a crucial for our algorithm. Figure 4.6a and Fig. 4.6b demonstrate annotations on unseen classes produced by our model when the background is both ignored and included during training. In each original image, the actual ground truth y is denoted as *GT*, while the annotation for seen classes y^s that the model sees prior to pseudo-labeling is represented by *labeled pixels*. Although our starting point (SPNet+ST) detects the presence of pixels of unseen classes, the predictions are noisy, with pixels assigned to classes that are not present in the current image. However, our consistency constraint (STRICT) reduces the noise, eliminating most of the pseudo-labels assigned to pixels of classes not present in the image (e.g. *train* in third row of Fig. 4.6a, *tv* in first and fourth rows of Fig. 4.6b). With more iterations, STRICT generates more refined pseudo-labels, where spatially coherent structures are present. This indicates that the pseudo-label generator captures global information of unseen classes, which is not possible to achieve with a single stage of pseudo-labeling.

4.4 Conclusion

This chapter focused on reducing the effort needed to collect and annotate datasets for learning novel classes over time. Two scenarios were explored: incremental Few-Shot Semantic Segmentation (iFSS), where only a few annotated images are available to learn new classes, and generalized zero-label semantic segmentation (GZLSS), where no annotation for unseen (novel) classes is provided during training.

To tackle iFSS, a formal definition was introduced and compared with existing settings we found in the literature. To address the challenges of the novel setting, PIFS was proposed, which combines prototype learning with knowledge distillation to achieve robust initialization of the parameters for the classifier on new classes and improve the network features representation. PIFS exploits prototypes of new classes as additional regularizers in the distillation loss to avoid overfitting and forgetting simultaneously. Furthermore, it also utilizes batch-renormalization to cope with non-*i.i.d.* data. An extensive benchmark showed that PIFS outperforms multiple incremental and few-shot methods that we adapted for iFSS.

For GZLSS, a self-training approach was proposed to segment classes not annotated in the training set by leveraging their semantic representation. The proposed method, STRICT, introduces a self-training pipeline that is simple, robust, and highly scalable. It relies on the model ability to predict consistent probabilities on different augmented versions of the same image to obtain coherent pseudo-labels. Futhermore, STRICT fine-tunes the model iteratively using the generated pseudo-labels, strengthening the performance of unseen classes over time. The effectiveness of this method was demonstrated on two commonly used benchmarks for semantic segmentation, outperforming other more complex strategies in the GZLSS.

We hope that our study will serve as a base for future research. In particular, we hope that our iFSS problem formulation and benchmark will bring forward the research to enable novel realistic and practical applications. In the future, on one hand, we aim to further expand this works in other segmentation tasks, such as instance and panoptic segmentation. On the other hand, we aim to study the property of multi-modal (image and language) models, such as CLIP [127], to extend their ability in zero-shot semantic segmentation.

Chapter 5

Weakly-Supervised Semantic Segmentation

5.1 Introduction

To train accurate semantic segmentation models it is necessary to collect pixel-level annotations for images in the dataset, a process that is time-consuming and expensive. The significant burden of requiring annotations for each pixel of an image has led to several research efforts toward building semantic segmentation models using cheaper, but weaker, annotation types. Under this perspective, different types of annotation have been explored, such as image-level labels [76, 60, 72, 150], bounding boxes [33, 117, 68], scribbles [82, 154] and points [11, 126]. A comparison of the annotation types and their cost is reported in Fig. 5.1. However, using weaker types of annotation requires a substantial effort to avoid performance drops.

In this chapter, we will first analyze how to train semantic segmentation model using point and scribbles annotations. We will extend the method present in Sec. 3.2 to model the unlabeled pixels in the images. In particular, when considering point and scribbles annotations, only a few pixels in the images are privded a label while the others are left unlabeled. We exploit the assumption that the image may only contain classes present in the annotation or the background and we adapt the cross-entropy loss presented in Sec. 3.2. We show that our simple method improves the state of the art on two point and one scribble benchmarks, outperforming the previous baselines on point supervision and achieving comparable results with ad-hoc method when using scribbles.

Additionally, we investigate how to extend semantic segmentation models with new classes over time using only image-level labels, proposing a novel framework, dubbed WILSON. It relies on a distillation framework to avoid forgetting old classes and it introduces an additional localizer module that extracts pixel-level pseudo-labels for the new classes. We introduce a novel Weakly Incremental Learning Semantic Segmentation (WILSS) benchmark to evaluate our method, showing it surprisingly achieves results comparable with methods using pixel-level supervision.

The work presented in this chapter led to the publication of two works:

- Cermelli, F., Mancini, M., Bulo, S. R., Ricci, E., and Caputo, B. *Modeling the background for incremental and weakly-supervised semantic segmentation*. In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.44, no.12, pp. 10099-10113 (2021).
- Cermelli, F., Fontanel, D., Tavera, A., Ciccone, M., and Caputo, B. *Incremental learning in semantic segmentation from image labels*. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4371-4381).

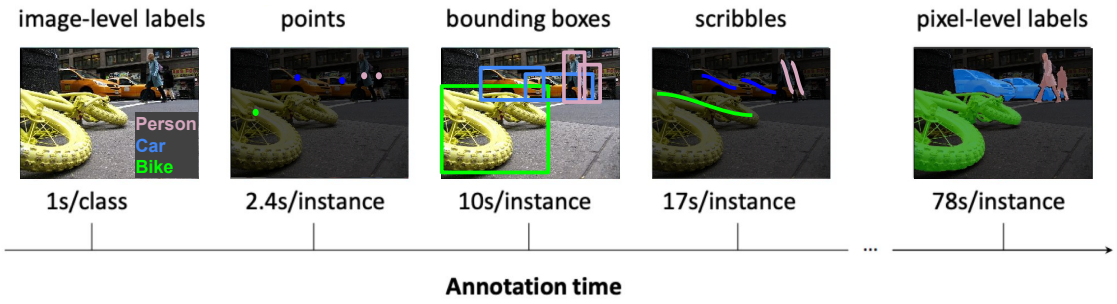


Fig. 5.1 Comparison of weakly annotation types. Time for annotation is taken from [11].

5.2 Semantic Segmentation from Point and Scribble Annotations

Point and scribble supervision can be a cost-effective solution for image annotation, but they result in partially annotated scenarios where only a few pixels are labeled with precise localization information for each instance of a class. All other pixels in the image, however, remain unlabeled, even if they may contain valuable information for learning to segment the classes. In particular, they may contain any of the weakly annotated classes in the image or the background. We report an example of this scenario in Fig.5.2 where there are point-level annotations for *bike*, *car* and *person* classes. The definition of the setting entails that non-annotated pixels may contain either the background or one of the three categories above but not other classes without annotations in the image (e.g., *train* and *bus*).

Previous works in point [11, 126] and scribble-supervised [82, 154] semantic segmentation disregard the information containted in the non-annoteted pixels. A common solution to deal with this scenario is to employ the partial cross-entropy, where the training loss is computed only on the annotated pixels, discarding from the optimization process all the others and thus losing information about the shape of the classes and wasting computation.

To address this issue and to exploit all the pixels in the image, we propose a loss, complementary to the partial cross-entropy, that operates on the unlabeled pixels. Similarly to incremental semantic segmentation (see Sec. 3.2), in this setting we consider all the non-annotated pixels as belonging to a fictious *background* class that shift its semantic from image to image, containing the real background or any of the annotated class. We encode this prior in the cross-entropy loss and, inspired by MiB (Sec. 3.2), we force the model to predict either a class present in the image or the background in the non-annotated pixels. We benchmark our approach in semantic segmentation in the Pascal-VOC dataset using both point [11] and scribble [82] supervision, showing performance superior or comparable to the

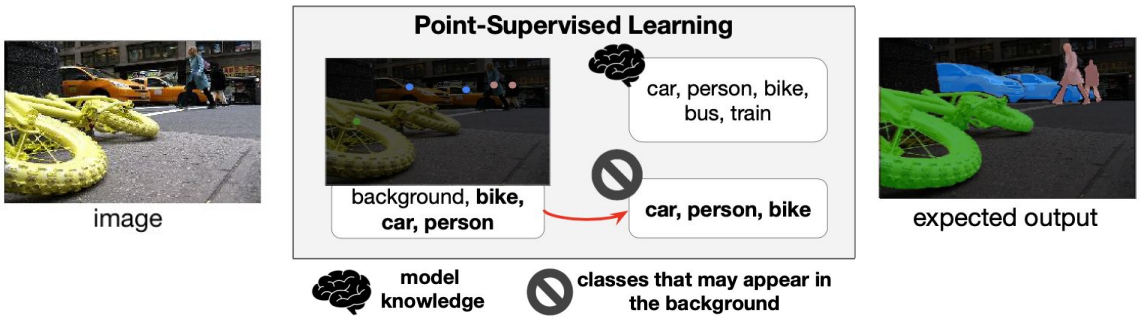


Fig. 5.2 In point-supervised learning, the annotations provides a few annotated pixels. All the other pixels are reported as background and they may contain any of the annotated classes.

state of the art. Furthermore, we consider a more challenging scenario, scene parsing with point supervision, where all the pixels should be assigned a semantic class. Experiments on ADE20k [126] demonstrate the effectiveness of our approach in this scenario.

To summarize, we provide the following contributions: (1) we propose to exploit the unlabeled pixels in the point and scribble supervision setting, extending the approach introduced in Sec. 3.2, and (2) we benchmark our novel approach on three weakly supervised setting, obtaining competitive results against previous specialized and complex baselines.

5.2.1 Related Works

Several research efforts have aimed to reduce the significant burden of requiring annotations for each pixel of an image in building semantic segmentation models. Cheaper, but weaker, types of annotation have been explored, such as image-level labels [76, 60, 72, 150], bounding boxes [33, 117, 68], scribbles [82, 154] and points [11, 126]. In this section, we focus on weak supervision using points and scribbles. Scribble annotations are a fast way to collect strong localization information for each class. In [82], the authors proposed to divide pixels into super-pixels and use pixel-similarity as additional source of supervision. In contrast, [154, 155] integrated graphical models into regularization losses to ensure consistent outputs on similar pixels. More recently, Wang et al. [171] proposed using two additional sub-networks to refine the model's output iteratively and predict boundaries for more precise segmentation results. Point supervision is more challenging, as it only provides one point for each instance in the image. In [11], the authors proposed a three-component approach, including an image-level prior to predict which objects are present in the image, a partial cross-entropy on the labeled points, and an objectness prior, extracted from an external model, to differentiate background and foreground pixels. In [126], the authors proposed a

method for scene parsing using point supervision, which included a partial cross-entropy and distance metric regularization to produce similar feature vectors for pixels of the same classes. Differently from previous works, we show that a simple loss formulation, which considers the uncertainty on unlabeled pixels, produces a performance improvement compared to the standard partial cross-entropy adopted by multiple previous works.

5.2.2 Semantic Segmentation using Weak Supervision

Problem Formulation The objective of weakly supervised segmentation with point or scribble annotations is to develop a model that can accurately predict the semantic class of each pixel in an image, similar to traditional semantic segmentation (refer to Section 2.1). However, unlike traditional semantic segmentation, the model is trained using a dataset with incomplete pixel-level annotations, consisting of only points or scribbles. Specifically, for each class instance in a training image, only one or a few annotated pixels that are contiguous are provided. Formally, for an image x containing a set of pixels $\mathscr{I}$ and its label y in the training set $\mathscr{T}$, the annotations are only provided for the pixels in $\mathscr{I}_S^x = \{i : \forall\, i \in \mathscr{I} \text{ s.t. } y_i \in \mathscr{Y}\}$, where $|\mathscr{I}_S^x| << |\mathscr{I}|$. All the other image pixels are unlabeled. In this study, we focus on three weakly semantic segmentation settings: point-based object segmentation [11], scribble-based object segmentation [82], and point-based scene parsing [126]. Object segmentation involves identifying and classifying countable objects, such as *cars*, *bikes*, and *dogs*, in an image. The remaining pixels are labeled as background (represented by `b`), which is considered a separate class in the model output space $\mathscr{Y}$. Formally, the model aims to predict a label $y_i \in \mathscr{Y}$ for each pixel i in a given image based on a training set $\mathscr{T} \subset \mathscr{X} \times (\mathscr{Y} \cup \mathtt{u})^N$, where `u` indicates unlabeled pixels and N is the total number of pixels ($N = |\mathscr{I}|$). Point annotations are provided only for the objects in the image, with no points given for the background class, following the protocols established in [11] and [82]. Differently, scribble-based object segmentation, as described in [82], includes annotations for the background class.

In contrast, scene parsing is a more complex task that involves predicting both countable objects and non-countable *stuff* classes such as *sky*, *road*, and *ground*. All pixels in the image are labeled with a semantic category, and the background class is not included in the label space. The aim is to learn a model that maps each pixel i in an image to a label $y_i \in \mathscr{Y}$ based on a training set $\mathscr{T} \subset \mathscr{X} \times (\mathscr{Y} \cup \mathtt{u})^N$.

Partial Cross-Entropy. As the amount of labeled pixels is limited, previous studies [11, 126] have proposed to use a cross-entropy loss on the annotated points. Specifically, they

introduced a partial cross-entropy (PCE) loss that only takes into account the pixels for which an annotation is provided. Given an image x with the corresponding annotation y, the PCE loss can be defined as:

$$\ell_{PCE}(x,y) = -\frac{1}{|\mathscr{I}_S^x|} \sum_{i \in \mathscr{I}_S^x} \log q_x(i, y_i), \tag{5.1}$$

with $q_x(i,c)$ indicating the probability predicted by the model for class c in pixel i.

The cross-entropy loss is crucial for enabling the network to differentiate between classes and accurately locate them within an image. Nevertheless, while using a PCE loss is simple and convenient, it disregards any information that may be obtained from the unlabeled pixels. To make use of this information, we will modify and adapt the principle introduced in Sec. 3.2 for the revised cross-entropy loss for this scenario.

Modeling the unlabeled. Our approach begins with the assumption that there is at least one labeled pixel for each instance of a class within an image and that all pixels in it must belong to one of these classes. By making use of this assumption, we can extract valuable information from the unlabeled pixels by maximizing the probability extracted from the model of having either one of these classes or the background. Formally, denoting the set of classes appearing in the label y of an image x as $\mathscr{U}_x = \{c : \exists i \in \mathscr{I}_S^x \text{ s.t. } c = y_i\}$ and $\mathscr{I}_u^x = \mathscr{I} \setminus \mathscr{I}_S^x$ the set of unlabeled pixels, we minimize the following loss function:

$$\ell_{UNL}(x,y) = -\frac{1}{|\mathscr{I}_u^x|} \sum_{i \in \mathscr{I}_u^x} \log p_x(i,u)\,, \tag{5.2}$$

where y_i is the ground truth label associated to pixel i and p_x is computed as follow:

$$p_x(i,c) = \begin{cases} q_x(i,c) & \text{if } c \neq \mathtt{u} \\ \sum_{k \in \mathscr{U}_x} q_x(i,k) & \text{if } c = \mathtt{u} \end{cases} \tag{5.3}$$

with $q_x(i,c)$ indicating the probability predicted by the model for class c in pixel i. Incorporating the ℓ_{UNL} loss into our training procedure offers two distinct advantages when combined with the ℓ_{PCE} loss. Firstly, the information from labeled pixels is propagated to the unlabeled pixels, resulting in an additional source of supervision. This enables the network to learn more effectively and make better use of the available labeled data. Secondly, in the event that the network predicts an unlabeled pixel as belonging to a class c that is not present in the current image (i.e., $c \notin \mathscr{U}_x$), the loss function provides feedback on the error. This is beneficial as it enables the network to learn from errors and generalize more effectively.

To summarize, given a training set $\mathscr{T}$, we train the network to minimize the following objective function:

$$\mathscr{L}(\theta) = \frac{1}{|\mathscr{T}|} \sum_{(x,y)\in\mathscr{T}} \ell_{PCE}(x,y) + \lambda\ \ell_{UNL}(x,y), \tag{5.4}$$

where λ is a hyperparameter that weight the importance of unlabeled pixels We recall that unlabeled pixels are many more than the labeled ones in weakly-supervised learning, thus we need this parameter to rebalance their contributions.

We note that the proposed loss function can be used for both point and scribble, as well as for both object segmentation and scene parsing tasks, without any modification. The only difference lies in the set of classes present in the image, where for object segmentation, the background class is always included, whereas for scene parsing, it is not considered. To the best of our knowledge, our approach is the first to achieve state-of-the-art results in both object segmentation and scene parsing using point and scribble supervision without the need for any other prior knowledge or additional data, such as objectness prior [11].

5.2.3 Experiments

Point-based Object Segmentation on Pascal-VOC To evaluate our method for object segmentation, we use the Pascal-VOC [43] dataset with the point annotations provided by [11]. We use a Resnet-101 [55] backbone with dilated convolutions, as in standard state-of-the-art architectures [27], and add a bilinear interpolation layer on top to recover the input resolution without introducing additional trainable parameters. As in as in [11], we initialize the backbone with an ImageNet pretrained model from [138], but we do not initialize the classifier due to inability to establish the correct mapping among the ImageNet indices published by [11] and the ImageNet classes. We re-implemented [11] using our same backbone and training protocol to guarantee fair comparison. We train the network using SGD with momentum 0.9, weight decay 10^{-4}, and a polynomial learning rate policy $base\_lr \cdot (1 - \frac{iteration}{max\_iterations})^{0.9}$. The initial learning rate is set to 10^{-5} for the methods in [11] and 10^{-4} for the fully-supervised baseline. We report the results for our method with both learning rates. We train with a batch size of 24 for 30 epochs, cropping the images to 512×512 and applying data augmentation as in [27].

We present the results of our experiments in Tab. 5.1, which includes the mean intersection over union (mIoU) and overall pixel accuracy (*P-Acc*). The first three rows of the table refer to the methods proposed in [11]. The *Img Lvl* method, as described in [11], is trained using

Table 5.1 Results on point-based weakly supervised segmentation on Pascal-VOC (mIoU in %).

Method	**mIoU**	**P-Acc**
Img Lvl [11]	33.2	76.0
Img Lvl + PCE [11]	34.7	58.9
Img Lvl + PCE + Obj [11]	42.1	81.5
PCE + bkg	38.8	81.9
Ours (lr 10^{-5})	45.3	82.3
Ours (lr 10^{-4})	**46.7**	**83.6**
Full Supervision	58.8	89.9

only image-level labels without considering the points location. This method achieves an mIoU of 33.2%, which is 4.4% better than the one reported in [11]. In the second row, we add the partial cross-entropy (PCE) loss as proposed in [11], and refer to it as *Img Lvl + PCE*. For this method, we use all the points available in the annotation without weighting them ($\alpha_i = 1, \forall i \in \mathscr{I}_S$). The addition of PCE leads to a 1.5% improvement in mIoU, but a 17.1% decrease in pixel accuracy. This can be attributed to the model bias towards semantic classes, which causes it to assign object labels even to background pixels. When introducing the Objectness Prior (*Img Lvl + PCE + Obj*), computed on an additional dataset following [11], the results further improve, with the method achieving an mIoU of 42.1% and a pixel accuracy of 81.5%.

Our method demonstrates superior performance compared to all three variants of [11]. We report our method twice in the comparison: *Ours (lr* 10^{-5}*)* uses the same learning rate as [11], while *Ours (lr* 10^{-4}*)* uses a learning rate of 10^{-4} which we found to be better. With both learning rates, our method achieves better performance than [11], indicating that our method is better at modeling unknown pixels. *Ours (lr* 10^{-4}*)* achieves an mIoU of 46.7% and a pixel accuracy of 83.6%, which is inferior to the fully supervised baselines of 12.1% and 6.3%, respectively. Notably, our method does not use any objectness prior computed on external data, which is a key difference from [11].

To further prove that the improvement of our method is due to the way we model unlabeled pixels rather than rescaling the contribution of the background class, we introduce a baseline referred to as *PCE + bkg*. In this method, we still use Eq. (5.4), but we only consider the background as a possible class for the unlabeled pixels. However, as shown in Tab. 5.1, this method fails to learn the classes properly, achieving an mIoU of 38.8%, which is 7.9% lower than *Ours (lr* 10^{-4}*)*. This is because considering all the unlabeled pixels as background (*PCE + bkg*) biases the model towards this class. In contrast, our method models the unlabeled pixels using the prior given by the point labels, which forces the network to predict them either as background or as any of the annotated classes.

Table 5.2 Results on scribble-based weakly supervised segmentation on Pascal-VOC (mIoU in %).

Method	**wo/ CRF**	**w/ CRF**
PCE	69.5	72.8
Ours	**72.3**	**75.1**
Scribble-Sup [82]	-	63.1
NormalizedCut [154]	72.8	74.5
KernelCut [155]	73.0	75.0
BPG [171]	**73.2**	**76.0**
Full Supervision	75.8	76.4

Scribble-based Object Segmentation on Pascal-VOC In this study, our method for scribble-supervised object segmentation was evaluated following the experimental protocol defined in [155, 171]. The Pascal-VOC [43] dataset and the scribble annotation released by [82] were used. The Deeplab-v2 architecture [26] with the Resnet-101 backbone [55] was employed, and dilated convolutions were used to obtain an output resolution 8 times smaller than the input, as in [155, 171]. Training the network was done on a single-scale resolution using a polynomial learning rate policy $base\_lr \cdot (1 - \frac{iteration}{max\_iterations})^{0.9}$, with a batch size of 10 images and the following hyperparameters: $base\_lr = 2.5 \cdot 10^{-4}$, momentum 0.9 and weight decay $5 \cdot 10^{-4}$. The network was trained for 20K iterations using 321×321 cropped images, which were horizontally flipped (left-right) and randomly scaled (from 0.5 to 2.0). During testing, we followed the approach used in previous works [155, 171] by using multi-scale inputs (*i.e.* [0.5, 0.75, 1.0, 1.25, 1.5]) and applying max voting to get the final prediction.

The mIoU with and without the dense CRF post-processing using scribble-supervision is reported in Tab. 5.2. The top part of the table presents the results of methods not explicitly designed for the scribble annotation, namely the PCE baseline and our method. The following part presents the scribble-specific state-of-the-art approaches [82, 154, 154, 171], and the fully-supervised upper-bound. Similarly to point supervision, the PCE baseline trains the network using the cross-entropy only on labeled pixels, as described in Eq. (5.1). The competitive performance of PCE, indicated by the mIoU of 72.8%, which is only 3.6% below the fully-supervised upper-bound of 76.4%, demonstrates that the model can extract meaningful information even from few pixels. However, by introducing our loss as reported in Eq. (5.4), we outperform the PCE baseline. Specifically, our method achieves 72.3% (+1.8% compared to PCE) without CRF and 75.1% (+2.3%) with CRF. This highlights the importance of utilizing unlabeled pixels to improve the results.

Our method achieves competitive performance compared to state-of-the-art methods. Specifically, when compared to NormalizedCut [154] and KernelCut [155], our method performs slightly worse without using the CRF but outperforms them while using it (+0.6% with

Table 5.3 Results on point-based weakly supervised scene parsing on ADE20K (mIoU in %).

	Our protocol		[126] protocol	
Method	**mIoU**	**P-Acc**	**mIoU**	**P-Acc**
PCE	22.4	60.9	20.2 (17.7)	58.3 (58.0)
PDML [126]	21.1	56.6	19.3 (19.6)	55.5 (61.0)
Ours	**22.9**	**62.2**	**21.0**	**59.5**
Full Supervision	29.7	68.8	25.1	66.0

respect to NormalizedCut and +0.1% with respect to KernelCut). We argue that NormalizedCut and KernelCut outperform our method without CRF because they already integrate the CRF in their training objective to better model the object boundaries. However, introducing CRF post-processing at inference improves our method's performance, recovering precise object boundaries, while having less impact on NormalizedCut and KernelCut. Finally, BPG [171] performs better than our method both without (+0.9%) and with (+0.9%) CRF post-processing. However, we note that BPG introduces two sub-networks in the segmentation architecture to model object boundaries, significantly increasing the number of parameters and requiring additional supervision for boundary prediction. In contrast, we propose a general method that introduces only a loss function on unlabeled pixels, without requiring any modification to the network architecture or additional supervision.

Scene Parsing on ADE20K We tested our method on the scene parsing task proposed by [126]. This task is based on the ADE20K dataset [202] and the point annotation used in the LID Challenge 2020 [1]. As the code from [126] was not publicly available, we re-implemented their method using the algorithm and details provided in their paper. Furthermore, we evaluated the method using two different training protocols since we found that the protocol proposed by [126] was sub-optimal. Both protocols use a Resnet-101 [55] architecture with dilated convolutions, followed by a bilinear interpolation layer to recover the input resolution. The first protocol is the one described in [126]. The network is trained using stochastic gradient descent (SGD) with momentum 0.9, weight decay 5×10^{-4}, and an initial learning rate of 2.5×10^{-4}. The learning rate is decayed using a polynomial schedule of $base\_lr \cdot (1 - \frac{iteration}{max\_iterations})^{0.8}$. We iterate over the dataset using a batch size of 16, and the images are randomly cropped to size 321×321. However, since the number of epochs was not specified in [126], we train the network for 60 epochs. The second protocol we follow is the one employed above for point-based object segmentation with the only difference that for this scenario we set the base learning rate to 10^{-3}.

[1] https://lidchallenge.github.io/challenge.html, track 2

Table 5.3 presents the mean Intersection-over-Union (mIoU) and overall pixel accuracy (P-Acc) results. The numbers reported in [126] are shown in brackets. To replicate the PCE baseline proposed in [126], we apply cross-entropy loss only on the pixels that have a labeled ground truth, as described in Eq.5.1. This baseline yields strong results: 22.4% mIoU with our protocol and 20.2% mIoU with the protocol of [126]. Notably, our results surpass those in [126] by 2.5% mIoU and 0.3% pixel accuracy. The PDML [126] baseline performs comparably to [126] but exhibits a drop in performance of 1.3% and 0.9% mIoU when compared to the PCE baseline. Our proposed method, however, outperforms both baselines with 22.9% mIoU using our protocol, a 0.5% improvement over PCE, and 21.0% using the [126] protocol, with a gap of 0.8% compared to PCE.

5.3 Incremental Learning from Image-Level Labels

The current incremental learning methods can learn new classes over time, but they necessitate costly annotations at the pixel level for training, which can be prohibitively expensive and restrict their applicability in real-world scenarios. In this section, we hypotesizes that, to learn new classes it may be feasible to transfer the model segmentation ability from old classes to new ones without needing pixel-level annotations. Ideally, we would extend segmentation models to new classes relying only on the cheapest form of annotation, image-level labels, which are widely accessible and easy to collect online.

Therefore, we present a new task called Weakly-Supervised Incremental Learning for Semantic Segmentation (WILSS). This innovative setup combines the properties of incremental learning (training solely on new class data) and weak supervision (inexpensive and widely available annotations). An illustration of WILSS is shown in Fig. 5.3. Applying existing weakly-supervised methods directly to incremental segmentation necessitates (i) extracting pixel-wise pseudo-supervision offline using a weakly-supervised approach [8, 175, 2, 150, 77] and (ii) updating the segmentation network by employing an incremental learning technique [21, 40, 101]. Nonetheless, we contend that generating pseudo-labels offline in incremental settings is sub-optimal, as it involves two separate training stages and disregards the model knowledge of prior classes, which can be leveraged to learn new classes more effectively.

We propose a weakly incremental learning framework for semantic segmentation, dubbed WILSON. This framework incrementally trains a segmentation model using online pseudo-supervision generated from image-level annotations, exploiting prior knowledge, to learn new classes. To extend the standard encoder-decoder segmentation architecture [26–28], we introduce a *localizer* module upon the encoder. The localizer serves to produce pseudo-supervision for the segmentation backbone. To transfer the segmentation ability from old classes to the new one and thus improve the pseudo-supervision, we train the localizer with a pixel-wise loss guided by the predictions of the segmentation model. This regularization fullfil two purposes. Firstly, it acts as a strong prior for the previous class distribution, informing the model on where old classes are located in the image. Secondly, it provides a saliency prior for extracting better object boundaries. In addition, we avoid the use of hard pseudo-labels, as commonly used in previous works [175, 8, 77], since they would introduce noise in the training process that is harmful for the perforance. Differently, we make use of soft-labels extracted from the localizer considering the class probability distribution assigned to each pixel in the image.

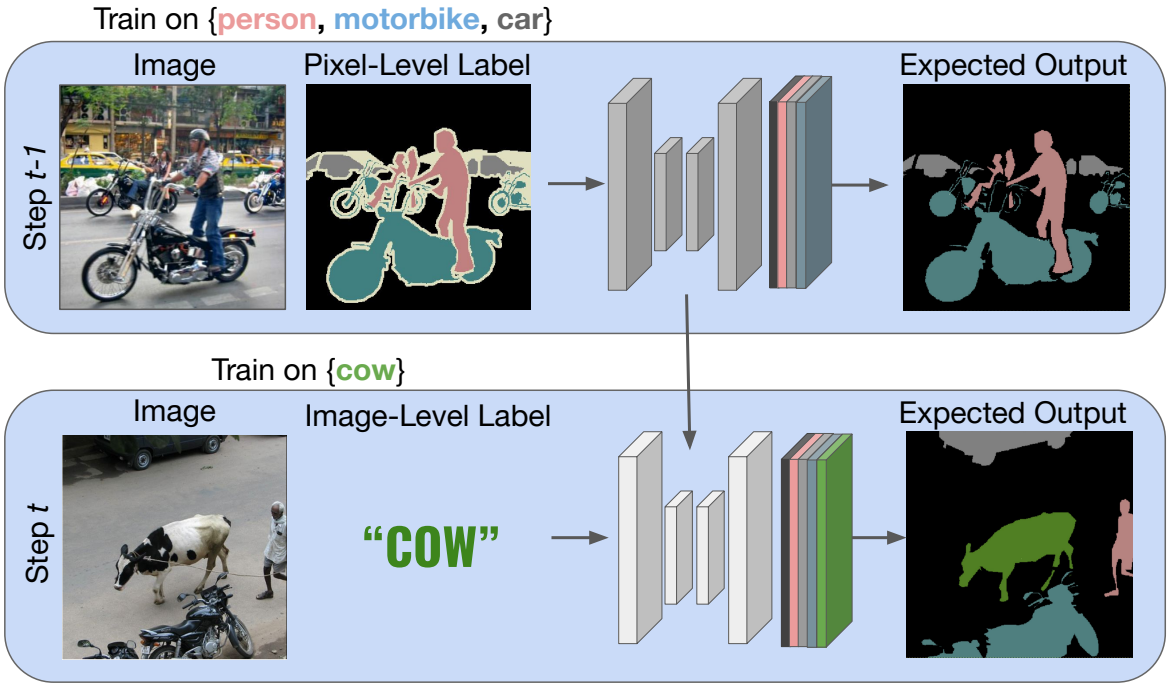


Fig. 5.3 Illustration of WILSS. A model is first pre-trained on a set of classes (*e.g.* , *person, motorbike, car*) using pixel-wise annotations. Then, the model is updated to segment new classes (*e.g.* , *cow*) exploiting image-level labels and without access to old data.

To summarize, the contributions are the following. (1) We propose the novel WILSS task with the aim of extending segmentation models with new classes using only image-level supervisio. (2) We propose a novel framework, WILSON, that generates soft pseudo-labels using a localizer from the image-level labels. (3) We evaluate WILSON on the Pascal VOC [43] and COCO [84] datasets, demonstrating that it surpasses offline weakly-supervised methods and achives comparable results w.r.t. pixel-supervised incremental learning methods.

5.3.1 Related work

Image-level supervision has become popular due to its low cost and high availability on the internet, receiving the most attention over other types of weak supervision. The majority of image-based weakly supervised approaches use a two-stage process [72, 111, 60, 2, 3, 76, 150, 22]. Firstly, they generate pixel-wise pseudo-labels by using a classification network, and then they use them to train a segmentation model. The pseudo-labels are typically obtained from the classification network by exploiting the Class Activation Maps (CAMs) [201]. Differently, Araslanov et al. [8] propose a one-stage approach in which a segmentation model is learned by generating pseudo-labels on the fly that are in turn used to self-supervise

the model. To improve the quality of the pseudo-labels, many researchers have introduced refinements steps [2, 3], additional losses [72, 60, 22, 175, 150, 8], or erasing techniques that force the CAM to focus on non-discriminative parts of the image [178, 58, 23]. Recently, some researchers have focused on using external information such as saliency estimation to improve the pseudo-labels on the object boundaries [77, 189]. Despite the advancements in pseudo-label generation techniques from image-level supervision, they operate in a static scenario where the model learns from a fixed set of classes. In contrast, we focus on the more challenging incremental learning setting where the model extends a trained segmentation model using only image-level labels.

5.3.2 WILSON Framework

In the following, we formally define the novel problem setting. Next, we illustrate the training procedure of the localizer, that has the goal of obtaining pseudo-supervision using image-level labels and the information coming from the segmentation model. Finally, we describe how to train the segmentation model to learn new classes without forgetting old ones. The overall framework is depicted in Fig. 5.4.

Problem Definition WILSS extends the incremental segmentation setting defined in Sec. 3.2, where training is realized over multiple *learning steps* and each learning step t introduces a novel set of classes $\mathscr{C}^t$ that, merged with the previous label set $\mathscr{Y}^{t-1}$, led to a new label set $\mathscr{Y}^t = \mathscr{Y}^{t-1} \cup \mathscr{C}^t$. In the initial step ($t = 0$) of WILSS, the model is trained on a dataset with pixel-level annotations only for the initial classes, *i.e.* $\mathscr{T}^0 \subset \mathscr{X} \times (\mathscr{C}^0)^N$. Then, in the following steps, the dataset contains only cheap image-level labels for the new classes. Specifically, for ($t > 0$), the model is provided a dataset with only image-level annotations for new classes $\mathscr{T}^t \subset \mathscr{X} \times (\mathscr{C}^t)$. As in standard incremental learning, we assume that, at step t, only the dataset containing new classes can be used, and all the previous datasets are not accessible anymore. The goal of WILSS is to obtain a model f_{θ^t} able to predict all the seen classes, such that $f_{\theta^t} : \mathscr{X} \mapsto \mathbb{R}^{N \times |\mathscr{Y}^t|}$. In the following, we assume the model is made by two components, an encoder e and a decoder d, such that $f(x) = d(e(x))$.

Training the Localizer Drawing from the literature on Weakly Supervised Semantic Segmentation (WSSS) [8, 175, 77, 80, 2, 72], we propose the use of a *localizer* g to generate pseudo-supervision for the segmentation model using image-level labels. The localizer utilizes the features of the segmentation encoder e to generate scores for all classes including the background, old, and new ones as follows: $z = g(e(x)) \in \mathbb{R}^{|\mathscr{Y}^t| \times N}$.

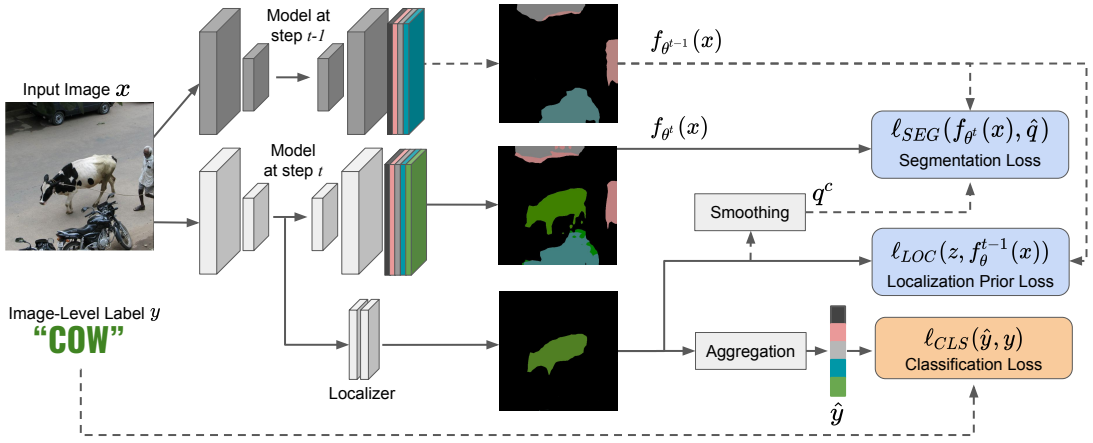


Fig. 5.4 Illustration of the end-to-end training of WILSON. The localizer is directly trained using a classification loss ℓ_{CLS} and the Localization Prior loss ℓ_{LOC}, which exploits the prior information of the old model at step $t-1$. The segmentation model is supervised using CAM and old model output. The gradient is not backpropagated on dotted lines.

In order to learn from image-level labels, we must first aggregate the pixel-level classification scores z. Typically, Global Average Pooling (GAP), *i.e.* averaging equally the scores from all the pixels, is used for this purpose [2, 175]. However, this method produces coarse pseudo-labels [8] because it uniformly encourages all pixels in the feature map to be discriminative for the target class. To obtain more precise pseudo-labels, we use the *normalized Global Weighted Pooling* (nGWP) method [8], which weights each pixel based on its relevance for the target class. Specifically, the weight of each pixel is calculated by normalizing the classification scores with the softmax operation ψ, i.e., $m = \psi(z)$ The image-level scores are computed as:

$$\hat{y}^{nGWP} = \frac{\sum_{i \in \mathscr{I}} m_i z_i}{\varepsilon + \sum_{i \in \mathscr{I}} m_i}, \tag{5.5}$$

where ε is a small value that avoids numerical instability. In order to incentivize the detection of all visible parts of the object, we introduce the *focal penalty* term as in [8]. It is computed as follow:

$$\hat{y}^{FOC} = (1 - \frac{\sum_{i \in \mathscr{I}} m_i}{N})^{\gamma} log(\lambda + \frac{\sum_{i \in \mathscr{I}} m_i}{N}), \tag{5.6}$$

where λ and γ are hyper-parameters and $N = |\mathscr{I}|$. We refer the readers to [8] for more details on the nGWP and the focal penalty.

After obtaining the image-level score, we can then train the localizer using image-level labels. We recall that, as in previous incremental learning scenario (see Sec. 3.2), we assume that only to image-level annotations y for the *new classes* $\mathscr{C}^t$ are provided in the dataset. We

use the *multi-label soft-margin loss* to train the localizer:

$$\ell_{CLS}(\hat{y},y) = -\frac{1}{|\mathscr{K}|}\sum_{c\in\mathscr{K}} y^c log(\hat{y}^c) + (1-y^c) log(1-\hat{y}^c), \tag{5.7}$$

where $\mathscr{K} = \mathscr{C}^t$, $\hat{y} = \sigma(\hat{y}^{nGWP} + \hat{y}^{FOC})$, and σ the logistic function. It can be noted that, while the loss is computed using annotations only for new classes, it is indirectly influencend by the scores of the old classes due to the softmax normalization in Eq. (5.5). However, since image-level annotations are cost-effective and new images can be annotated with ease, it is possible to relax the conditions and allow for weak annotations for both old and new classes. Under this scenario, the classification loss in Eq. (5.7) is calculated for all classes, and $\mathscr{K}$ is equivalent to the set of all target classes $\mathscr{Y}^t$.

Localization Prior. The supervision provided by image-level labels is limited to the presence of new classes in the image, and does not indicate their boundaries or the location of old classes. However, we propose that such information can be gleaned from a previously learned segmentation model. Specifically, the background score from the segmentation model can serve as a saliency prior for improved object boundary extraction. Additionally, the scores of old classes can guide the localizer in identifying and localizing their presence in the image, thus directing its focus to alternate regions. Hence, we design a regularization loss that aim at providing direct supervision on the localizer from the segmentation model trained on step $t-1$, *i.e.* f_θ^{t-1}. We consider the regularization as a *Localization Prior* (LOC). It is computed as a pixel-wise loss between the segmentation model outputs $\omega = \sigma(f_\theta^{t-1}(x))$ and the classification scores z:

$$\ell_{LOC}(z,\omega) = -\frac{1}{|\mathscr{Y}^{t-1}|N}\sum_{i\in\mathscr{I}}\sum_{c\in\mathscr{Y}^{t-1}} \omega_i^c log(\sigma(z_i^c)) + (1-\omega_i^c) log(1-\sigma(z_i^c)), \tag{5.8}$$

where σ represents the logistic function. By incorporating the loss, the segmentation model generates a pixel-level objective on previous categories, thereby transferring its segmentation capability to the localizer. As opposed to the softmax operator that imposes rivalry between categories, using the logistic function decouples the likelihood of each class, which is advantageous to obtain accurate localization prior. Whenever a new class is present, the old categories and the background will have a reduced score, indicating to the localizer that the pixel pertains to a new class.

Soft Pseudo-Labels. To train the semantic segmentation model, standard WSSS methods commonly extract hard-pseudo labels from the image-level classifier. Specifically, they obtain a one-hot distribution $q^{\mathrm{H},c}$ for each pixel, assigning a value equals one to the class with

the maximum classification score and zero to the other classes. Formally, they are obtained as follows:

$$q_i^{\mathrm{H},c} = \begin{cases} 1 & \text{if } c = \arg\max_{k \in \mathscr{Y}^t} m_i^c, \\ 0 & \text{otherwise,} \end{cases} \tag{5.9}$$

with m representing the softmax normalized score extracted from the localizer.

Despite being commonly employed in previous works, it is well-known that hard pseudo-labels generated from an image-level classifier are noisy [80, 77, 8, 175]. Directly using $q^{H,c}$ tfor supervising the segmentation network might harm the learning process, causing the model to fit incorrect targets and leading to poor performance. To reduce the impact of the noise, we propose the use of a smoothing operation [97] to generate soft-pseudo labels. Formally, given a class c, the pseudo-supervision q^c is computed as:

$$q^c = \alpha q^{\mathrm{H},c} + (1-\alpha) m^c, \tag{5.10}$$

where α is a hyperparameter that controls the smoothness in the pseudo-labels.

The localizer produces scores for both old and new classes, but the pseudo-labels output distribution may be strongly biased towards new classes, since the novel dataset mainly includes images depicting them. Consequently, adopting q as the only target for the segmentation model would result in catastrophic forgetting [102]. To address this issue, we draw inspiration from the knowledge distillation framework [57, 81] and replace the pseudo-supervision derived from the localizer on old classes with the output of the segmentation model f_θ^{t-1} that was trained in the preceding learning phase. The final pixel-level pseudo-supervision $\hat{q}$ is thus composed as follows:

$$\hat{q}^c = \begin{cases} \min(\sigma(f_{\theta^{t-1}}(x))^c, q^c) & \text{if } c = \mathtt{b}, \\ q^c & \text{if } c \in \mathscr{C}^t, \\ \sigma(f_{\theta^{t-1}}(x))^c & \text{otherwise,} \end{cases} \tag{5.11}$$

where $\mathtt{b}$ is the background class and σ is the logistic function. We note that, to alleviate the background shift issue (see Sec. 3.2), we utilize the minimum value of the two distributions for the background class.

Learning to Segment from Pseudo-Supervision. The pseudo-supervision $\hat{q}^c$ is composed by indepentent class probability scores. For this reason, we avoid the use of a standard softmax-based cross-entropy loss and we propose the use of the multi-label soft-margin loss

to train the segmentation model. Specifically, the segmentation loss is computed as follow:

$$\ell_{SEG}(p,\hat{q}) = -\frac{1}{N}\sum_{i\in\mathscr{I}}\sum_{c\in\mathscr{Y}^t}\hat{q}_i^c log(\sigma(f_{\theta^t}(x)_i^c)) + (1-\hat{q}_i^c)log(1-\sigma(f_{\theta^t}(x)_i^c)). \quad (5.12)$$

In conclusion, we remark that the localizer is discarded after the incremental learning step and it is not employed during the testing phase, thus it does not increase neither the time required for the inference nor the number of parameters.

5.3.3 Experiments

Datasets and Settings

We conducted a comprehensive evaluation of WILSON using two widely used benchmarks, Pascal VOC 2012 [43] and MS-COCO [84]. To benchmark our method, we followed the incremental semantic segmentation setting defined in Sec. 3.2 and evaluated it on two different settings using the Pascal VOC dataset. The first setting, **15-5 VOC**, involved learning 15 classes in the first phase and then adding 5 new classes in the second phase. The second setting, **10-10 VOC**, involved performing two steps of 10 classes each. We used two experimental protocols, namely the *disjoint* and *overlap* scenarios, to report the results. The *disjoint* scenario included images containing only new or previously seen classes in each training step, while the *overlap* scenario included all images containing at least one pixel from a novel class in each training step. We also introduced a novel incremental learning scenario, the **COCO-to-VOC**, consisting of two training steps. In the first step, we learned the 60 COCO classes that were not present in the Pascal VOC dataset, and we removed all images containing at least one pixel of the latter. In the second step, we learned 20 Pascal VOC classes. We reported the results on the validation sets as the test set labels were not publicly released. To evaluate the performance of the segmentation model, we used the mIoU. We remark that unlike the previous incremental learning setting, in the proposed WILSS setting, the incremental steps provided only image-level labels for the new classes.

Baselines Since WILSS is a novel setting, we lack direct comparable baselines. As a result, we evaluate WILSON against two categories of methods, namely pixel-supervised incremental learning approaches and weakly supervised semantic segmentation (WSSS) methods tailored for this setting. We report the results of eight pixel-supervised methods that represent the current state-of-the-art in incremental learning: LWF [81], LWF-MC [132], ILT [103], MiB [21] (described in Section 3.2), PLOP [40], CIL [71], SDR [104], and

RECALL [101]. It should be noted that RECALL [101] uses additional images from the Web, unlike other methods. For the Pascal VOC dataset, we use the results published in [101, 40], while for COCO-to-VOC, we run the experiments using the same code from Section 3.2. Furthermore, we evaluate the performance of several state-of-the-art WSSS methods, which we adapt to work in the incremental learning scenario. Specifically, we first train a classification model using the images available in the incremental learning steps. Then, we generate the hard pseudo-labels offline and train the segmentation model minimizing the loss in Equation 5.12. We report the results obtained with pseudo-labels generated from four methods: class activation maps (CAM), SEAM [175], SS [8], and EPS [77]. It is worth mentioning that EPS uses an off-the-shelf saliency detector trained on external data, while CAM, SS, and SEAM rely solely on image-level labels. All the baselines and WILSON were trained using the same experimental protocols. We used the implementation provided by the authors for each method to produce the results. To generate the pseudo-labels for CAM, we used the EPS implementation.

Implementation Details For all experiments, we utilize the Deeplab V3 architecture [26] with a ResNet-101 [55] backbone and output stride equal to 16 for Pascal VOC, and a Wide-ResNet-38 [181] with output stride 8 for COCO, both pre-trained on ImageNet. To decrease the memory footprint required by the experiments, we apply in-place activated batch normalization [138]. The localizer that generates the CAMs is composed of 3 convolutional layers, followed by batch normalization and Leaky ReLU, with the first two layers having a kernel size of 3×3, the last layer having a kernel size of 1×1, channel numbers of s {256, 256, number of classes}, and a stride of 1. We train the model for 40 epochs, using SGD with an initial learning rate of 0.001 (0.01 for the Deeplab head and the localizer), momentum 0.9, and weight decay 10^{-4}, and a batch size of 24. For the first 5 epochs, we only train the localizer. Afterward, we train the whole network by including pseudo-supervision from the localizer and decay the learning rate using a polynomial schedule with a power of 0.9. Similar to [8], we set $\lambda = 0.01$ and $\gamma = 3$ of Eq. (5.6). Following Eq. (5.6), we also use the self-supervised segmentation loss on the localizer after the fifth epoch. For all experiments, we set $\alpha = 0.5$ in Eq. (5.10).

Results

Single step addition of five classes (15-5). After the initial learning stage, the following 5 classes of the VOC dataset are added: *plant, sheep, sofa, train, and tv-monitor*. Results are reported in Tab. 5.4 of the paper. Despite being trained with only image-level labels,

Table 5.4 Results on the 15-5 setting of Pascal VOC expressed in mIoU%. The best method using Image-level supervision is bold. The best method using Pixel supervision is underlined. ⋆: results from [101]. ⋄: results from [40].

		Disjoint			Overlap		
Method	**Sup**	**1-15**	**16-20**	**All**	**1-15**	**16-20**	**All**
Joint ⋆	Pixel	75.5	73.5	75.4	75.5	73.5	75.4
FT ⋆	Pixel	8.4	33.5	14.4	12.5	36.9	18.3
LWF ⋆ [81]	Pixel	39.7	33.3	38.2	67.0	41.8	61.0
LWF-MC ⋆ [132]	Pixel	41.5	25.4	37.6	59.8	22.6	51.0
ILT ⋆ [103]	Pixel	31.5	25.1	30.0	69.0	46.4	63.6
CIL ⋆ [71]	Pixel	42.6	35.0	40.8	14.9	37.3	20.2
MIB ⋆ [21]	Pixel	71.8	43.3	64.7	75.5	49.4	69.0
PLOP ⋄ [40]	Pixel	71.0	42.8	64.3	<u>75.7</u>	51.7	<u>70.1</u>
SDR ⋆ [104]	Pixel	<u>73.5</u>	47.3	<u>67.2</u>	75.4	52.6	69.9
RECALL ⋆ [101]	Pixel	69.2	<u>52.9</u>	66.3	67.7	<u>54.3</u>	65.6
CAM	Image	69.3	26.1	59.4	69.9	25.6	59.7
SEAM [175]	Image	71.0	33.1	62.7	68.3	31.8	60.4
SS [8]	Image	71.6	26.0	61.5	72.2	27.5	62.1
EPS [77]	Image	72.4	38.5	65.2	69.4	34.5	62.1
WILSON (ours)	Image	**73.6**	**43.8**	**67.3**	**74.2**	**41.7**	**67.2**

WILSON achieves competitive results in both disjoint and overlap settings compared to approaches trained with pixel-wise supervision. In the disjoint scenario, WILSON overall outperforms RECALL and SDR by 1.0% and 0.1%, respectively, while maintaining enough plasticity for learning new classes without requiring a replay buffer. Additionally, WILSON surpasses PLOP and MIB by 1.0% and 0.5% on new classes. Comparing WILSON to WSSS competitors, the results demonstrate the strengths of WILSON, including its ability to retain knowledge of past classes and learn new semantic classes given only image-level annotations. For new classes, WILSON outperforms EPS by +5.3% mIoU in the disjoint scenario, despite the latter using saliency maps generated from an external off-the-shelf model. Furthermore, SEAM is outperformed by 11.7% and SS by 17.8%. In the overlap scenario, WILSON not only preserves all prior knowledge but also achieves a +7.2% boost when learning new classes compared to EPS, resulting in an overall improvement of +5.1% compared to the best methods (SS, EPS).

Single step addition of ten classes (10-10). In this setting, we introduce10 classes in the incremental step, namely: *dining-table, dog, horse, motorbike, person, plant, sheep, sofa, train, tv-monitor*. As shown in Tab. 5.5, our results are consistent with the 15-5 setting. The differences between our WILSON method and the IL (pixel-wise supervision) methods

Table 5.5 Results on the 10-10 setting of Pascal VOC expressed in mIoU%. The best method using Image-level supervision is bold. The best method using Pixel supervision is underlined. ⋆:results from [101].

		Disjoint			Overlap		
Method	**Sup**	**1-10**	**11-20**	**All**	**1-10**	**11-20**	**All**
Joint ⋆	Pixel	76.6	74.0	75.4	76.6	74.0	*75.4*
FT ⋆	Pixel	7.7	60.8	33.0	7.8	58.9	32.1
LWF ⋆ [81]	Pixel	63.1	61.1	62.2	<u>70.7</u>	63.4	67.2
LWF-MC ⋆ [132]	Pixel	52.4	42.5	47.7	53.9	43.0	48.7
ILT ⋆ [103]	Pixel	<u>67.7</u>	<u>61.3</u>	<u>64.7</u>	70.3	61.9	66.3
CIL ⋆ [71]	Pixel	37.4	60.6	48.8	38.4	60.0	48.7
MIB ⋆ [21]	Pixel	66.9	57.5	62.4	70.4	63.7	67.2
PLOP [40]	Pixel	63.7	60.2	63.4	69.6	62.2	67.1
SDR ⋆ [104]	Pixel	67.5	57.9	62.9	70.5	<u>63.9</u>	<u>67.4</u>
RECALL ⋆ [101]	Pixel	64.1	56.9	61.9	66.0	58.8	63.7
CAM	Image	**65.4**	41.3	54.5	**70.8**	44.2	58.5
SEAM [175]	Image	65.1	53.5	60.6	67.5	55.4	62.7
SS [8]	Image	60.7	25.7	45.0	69.6	32.8	52.5
EPS [77]	Image	64.2	54.1	60.6	69.0	57.0	64.3
WILSON (ours)	Image	64.5	**54.3**	**60.8**	70.4	**57.1**	**65.0**

are minimal, and their results are nearly comparable. However, using the most accurate incremental learning method, ILT, the gap in accuracy is 3.9% in the disjoint scenario and shrinks to 2.4% in the overlap scenario when compared to SDR. Our technique outperforms all offline WSSS competitors in the overlap protocols by more than +0.7% overall mIoU, while achieving a comparable result (+0.2%) in the disjoint scenario. Qualitative results demonstrating the superiority of WILSON on both new and old classes are shown in Fig. 5.5.

COCO-to-VOC. The most challenging set of experiments involves training the network on 60 classes from the COCO dataset that are not shared with VOC, and then adding an additional 20 classes from the VOC dataset in a second step. Evaluation of this experiment is shown in Tab. 5.6 on both COCO and VOC validation sets. While WILSON's performance drops 8% compared to LwF when learning new classes, this experiment demonstrates our ability to retain prior information while learning new classes under image-level supervision, surpassing ILT performance on old classes (+2.8%), which is the top competitor trained with pixel-wise supervision. WILSON outperforms all previous weakly supervised methods on both old and new classes, both on COCO and VOC, with improvements of 4.8% in mIoU from the best WSSS method (EPS) on COCO.

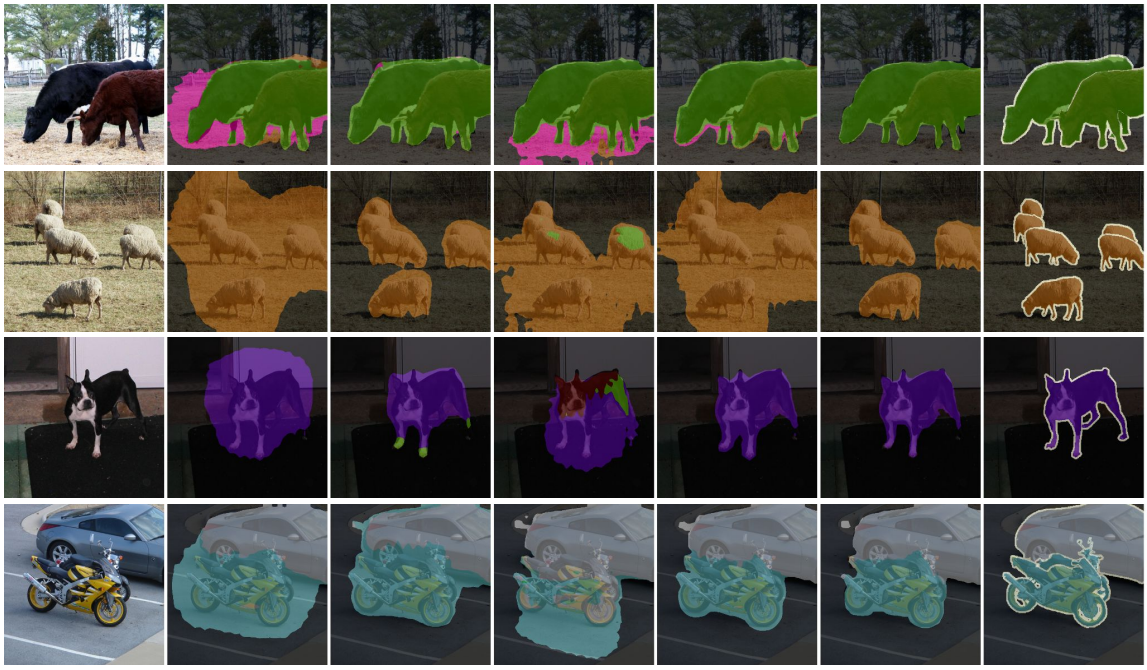


Fig. 5.5 Qualitative results on the 10-10 VOC setting comparing different weakly supervised semantic segmentation methods. The image emphasized the efficiency of WILSON in both learning new classes (e.g. sheep, dog, motorbike) and preserving knowledge of old ones (e.g. cow, car). From left to right: image, CAM, SEAM [175], SS [8], EPS [77], WILSON and the ground-truth. Best viewed in color.

Ablation Studies

Localization prior. In order to assess the effectiveness of the pseudo-supervision generation, we conducted an ablation study by exploring various choices for training the localizer. Results are reported in Tab. 5.8 for both the VOC 10-10 disjoint and overlap scenarios. The different training strategies that are compared include: (i) using a constant value for the old classes, as in [8]; (ii) using a fixed prior by concatenating the segmentation output of the old model to the class scores when calculating m; (iii) providing a localization supervision to the localizer using the softmax cross-entropy loss; and (iv) using the loss in Eq. (5.8). When a constant value was used and past knowledge from the old segmentation network was disregarded, it resulted in lower performance in comparison to the overall mIoU obtained when using a localization prior, particularly on new classes (-4.4% on disjoint and -5.1% on overlap). This indicates that teaching the localizer the location of previous classes is an effective way to prevent forgetting and improve performance when learning new classes. However, using aggressive priors, such as directly using the segmentation output of the old model, hindered the network's ability to effectively learn new classes, thus creating a gap of -4.0% on disjoint and -4.3% on overlap scenarios with respect to ℓ_{LOC}. Additionally, using the

Table 5.6 Results on the COCO-to-VOC setting expressed in mIoU%. The best method using Image-level supervision is bold. The best method using Pixel supervision is underlined.

		COCO			VOC
Method	**Sup**	**1-60**	**61-80**	**All**	**61-80**
FT	Pixel	1.9	41.7	12.7	<u>75.0</u>
LWF [81]	Pixel	36.7	<u>49.0</u>	<u>40.3</u>	73.6
ILT [103]	Pixel	<u>37.0</u>	43.9	39.3	68.7
MIB [21]	Pixel	34.9	47.8	38.7	73.2
PLOP [40]	Pixel	35.1	39.4	36.8	64.7
CAM	Image	30.7	20.3	28.1	39.1
SEAM [175]	Image	31.2	28.2	30.5	48.0
SS [8]	Image	35.1	36.9	35.5	52.4
EPS [77]	Image	34.9	38.4	35.8	55.3
WILSON (ours)	Image	**39.8**	**41.0**	**40.6**	**55.7**

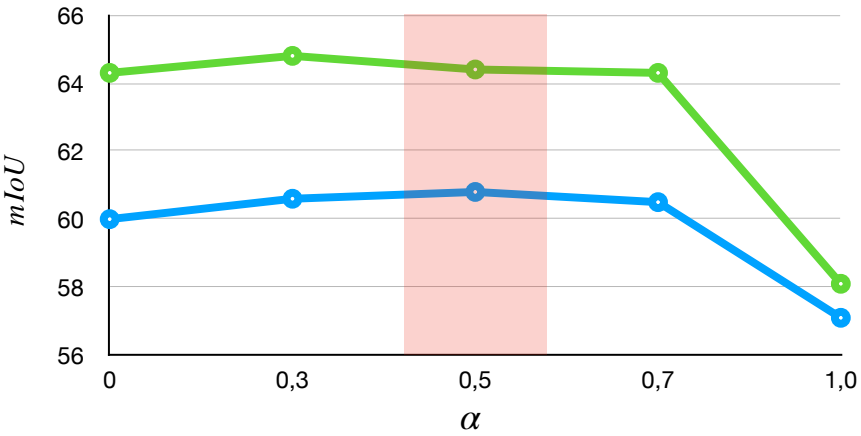


Fig. 5.6 Ablation study about the effect of α to smooth the one-hot pseudo-labels used to supervise the ℓ_{SEG}. Test reporting the mIoU for both the Disjoint and Overlap VOC 10-10 protocols.

softmax cross-entropy loss to match the segmentation output proved to be detrimental for performance, resulting in poor results for both new and old classes (-6.3% on disjoint and -5.8% on overlap with respect to ℓ_{LOC}). This can be attributed to the fact that the cross-entropy loss, due to softmax normalization, does not consider each class independently, and forces the localizer to produce high scores for old classes even when they have low segmentation scores.

Smoothing effect on pseudo-supervision. The smoothness of the pseudo-labels supervising the segmentation model is regulated by the hyper-parameter α of Eq. (5.8). To optimize the performance of the model, α should be properly tuned. We demonstrate that the model is robust to this choice. We report five different values of α, ranging from 0 to 1 and the final

Table 5.7 Performance evaluation of weakly supervised segmentation methods trained with direct supervision on both old and new classes in the incremental step.

	VOC 15-5					
	Disjoint			Overlap		
Method	**1-15**	**16-20**	**All**	**1-15**	**16-20**	**All**
CAM	70.5	34.7	62.6	71.6	36.0	63.7
SEAM [175]	71.9	26.9	61.7	70.8	28.1	61.0
SS [8]	71.8	26.3	61.7	72.1	27.6	62.1
EPS [77]	73.5	45.7	67.7	75.3	**47.6**	69.4
WILSON (ours)	**75.0**	**46.0**	**68.9**	**76.1**	45.6	**69.5**

	VOC 10-10					
	Disjoint			Overlap		
	1-10	**11-20**	**All**	**1-10**	**11-20**	**All**
CAM	63.1	42.2	53.9	66.6	45.0	56.8
SEAM [175]	66.0	50.4	59.7	70.9	54.6	64.0
SS [8]	60.8	26.0	45.2	69.6	33.0	52.6
EPS [77]	69.1	53.0	62.4	72.9	55.7	65.4
WILSON (ours)	**69.5**	**56.4**	**64.2**	**73.6**	**57.6**	**66.7**

Table 5.8 Ablation study to validate the robustness of pseudo-supervision considering different types of localization priors for training the localizer.

		Disjoint			Overlap		
Prior	**Loss**	**1-10**	**11-20**	**All**	**1-10**	**11-20**	**All**
-	-	64.8	49.9	58.8	69.4	52.0	62.0
Fixed	-	**66.1**	50.3	59.7	**71.4**	52.8	63.4
Learned	CE	61.1	46.0	54.5	67.6	49.5	59.2
Learned	ℓ_{LOC}	64.5	**54.3**	**60.8**	70.4	**57.1**	**65.0**

mean Intersection over Union (mIoU) in the VOC 10-10 disjoint and overlap scenarios in Fig. 5.6. The use of hard labels (i.e., $\alpha = 1$) resulted in the model fitting the noise in the supervision and forgetting prior knowledge, leading to poor performance and incapacity to learn novel classes. For our experiment, we chose $\alpha = 0.5$ as it balances between learning and remembering. We noted that changing the values of α from 0 to 0.7 only marginally affected the results, with an average difference of less than 0.5% between the disjoint and overlap case.

Using supervision for all the classes. In this experiment, the evaluation is conducted by providing image-level supervision for both old and new classes in incremental steps. The results on VOC are reported in Tab. 5.7. A comparison with Tab. 5.4 and Tab. 5.5 reveals a

notable improvement in performance. Specifically, all methods show an improvement, with WILSON exhibiting an average improvement of 2% on both old and new classes in the 15-5 and 10-10 settings. These findings highlight the importance of incorporating knowledge about old classes in pseudo-supervision generation for effective learning of new classes and avoidance of forgetting. Furthermore, the results demonstrate that WILSON outperforms offline WSSS methods in this scenario as well. WILSON achieves better performance in every setting, surpassing EPS by 1.2% and 0.1% in the VOC 15-5, and by 1.8% and 1.3% in the VOC 10-10 for the disjoint and overlapped scenarios, respectively.

5.4 Conclusion

This chapter presented two solutions for one of the crucial factors that limits the application of segmentation models: the annotation cost. We addressed the problem by investigating three different types of weak supervision: point, scribble, and image-level labels.

In the first part of the chapter, we proposed a general method aimed at learning using point or scribble supervision. We introduced a novel loss formulation that considered unlabeled pixels as ground-truth annotations for *any* possible class that the image might have contained, i.e., the classes with at least one annotated pixel. We benchmarked our novel loss function against specialized methods on either point or scribble supervision in three settings: point-based and scribble-based object segmentation using Pascal-VOC, and point-based scene parsing on the challenging ADE20K dataset. Our model obtained competitive performance with respect to previous approaches in both object segmentation and scene parsing despite being general to both tasks and without any additional prior on the objects or making assumptions on the provided annotations.

In the second part, we introduced a new setting called WILSS, which was designed to update the knowledge of semantic segmentation models by utilizing low-cost image-level annotations. Traditionally, weakly supervised learning techniques would require the creation of pseudo-supervision offline, followed by the training of the segmentation model. However, our proposed method, WILSON, took a different approach by coupling the semantic segmentation model with a localizer and utilizing image-level annotations on new classes to produce pseudo-supervision online for the segmentation network. The results indicated that the incorporation of a localization prior from the old model into the localizer significantly improved the generation of pseudo-labels. To test the efficacy of our approach, we performed three incremental learning experiments, and the results demonstrated that our method outperformed WSSS baselines and achieved results that were comparable to fully supervised incremental learning methods.

This chapter brings new solutions for an important issue in semantic segmentation. However, these works are only the first step towards obtaining segmentation models that were robust and that could learn incrementally new classes from heterogeneous weak annotations, effectively exploiting the large amount of datasets available on the web.

Chapter 6

Conclusions and Future Works

6.1 Summary of Contributions

During the thesis, we explored how to incrementally add novel classes to a semantic segmentation model, without forgetting the previous knowledge. This task introduces additional challenges with respect to the tasks investigated by previous works (e.g., image classification), due to the presence of multiple classes in each image in semantic segmentation. Furthermore, given the prohibitive cost for annotating images at the pixel-level, we focused on data-efficient techniques, that are able to learn novel classes, without forgetting, reducing the amound of data and annotations required for updating the model. Specifically, in Chapter 3, we investigated the challenges of incremental learning in semantic segmentation, extending the findings on object detection. Then, in Chapter 4, we focused on reducing the amount of images required for the training. Finally, in Chapter 5, we studied techniques able to learn a semantic segmentation model without using pixel-level supervision but cheap annotations such as point, scribbles or image-level labels.

Incremental Learning in Segmentation and Object Detection. In Chapter 3 we investigated incremental learning in complex vision tasks: semantic segmentation, object detection, and instance segmentation. We found that these tasks introduces additional challenges with respect to the simple classification due to the presence in each image of multiple classes that may belong to either new categories, or classes learned in the past, or that will be learned in the future, with only the novel classes being annotated. This characteristic exacerbates catastrophic forgetting, harming the performance of the model even after few incremental steps. We analyzed the problem in semantic segmentation, where the classes not present in the annotation are considered as the background, leading to the background-shift issue: at

every training step the semantic of the background in the training set changes, including all the classes outside the ones that are being trained. We proposed a method, dubbed MiB, that models the background shift by revisiting a standard knowledge distillation framework and effectively alleviates catastrophic forgetting. In addition, we showed that a similar problem is present in incremental learning for object detection, where old and future categories may be present but not annotated and thus not considered as an actual object in the training step. Taking inspiration from MiB, we proposed to model the missing annotations (MMA) by designing an approach that revisits the losses employed in common object detection frameworks.

Few-Shot and Zero-Label Semantic Segmentatation. We aimed in Chapter 4 to reduce the burden of collecting and annotating datasets, introducing two techniques able to learn new classes over time being provided only a few annotated images or even a simple textual description for them. For the former setting, we introduced the Incremental Few-Shot Semantic Segmentation scenario and we proposed PIFS, a method that combines prototype learning with knowledge distillation to improve classifier parameter initialization and network feature representation. PIFS employs prototypes of new classes as additional regularizers in the distillation loss to prevent overfitting and forgetting simultaneously. The latter setting considered is generalized zero-label semantic segmentation, where we proposed STRICT. It is a self-training approach that uses the model's ability to predict consistent probabilities on augmented images to generate coherent pseudo-labels for unseen classes. The method fine-tunes iteratively using these labels, improving its performance over time.

Weakly-Supervised Semantic Segmentatation Chapter 5 presented two solutions to address the high cost of annotating datasets in semantic segmentation models entirely avoiding the use of pixel-level labels. We proposed a general method to learn using point or scribble supervision. We designed an objective function that not only exploits the few labels pixels, but also considers the one unlabeled. Specifically,starting from the assumption that all the pixels in the image must contain one of the classes reported in the annotation, we design an loss that minimize the probability of having anyone of them in each pixel. This techniques matches the performance of complex and specific methods while being general to both tasks and without requiring external data. In the second part, we investigated a solution for incrementally add new classes to a pretrained semantic segmentation model using cheap and widely available image-level labels. We name this setting WILSS and we proposed a novel framework, WILSON, that couples a knowledge distillation framework for semantic segmentation with an additional localizer that is able to produce pseudo-supervision starting from image-level labels.

6.2 Open Issues and Future Works

In the following, we discuss some open issues of data-efficient incremental learning in segmentation, proposing future research direction of our work.

Benchmarks and Performance on long incremental learning tasks. Despite the rapid advances of the recent years, learning new classes over time without forgetting is still challenging task, especially when considering complex tasks such as semantic segmentation and object detection. Specifically, to truly design incremental learning methods able to operate in the real-world, it is important to benchmark them on very long sequences of tasks, where the model has to learn thousands of new classes during its lifetime. However, the performance on such settings are often poor since the model is prone to forget the learned knowledge after tens of tasks. In the future, it would be important to design more realistic benchmarks, closing the gap with real-world applications.

Exploting Multi-Modal Models for Incremental Learning. The majority of incremental learning methods rely on un-informative numerical labels to train their network. Very recently, however, new multi-modal vision and language models [127] have been developed. These models can predict classes being only provided captions or textual descriptors for them and they have very interesing performance when tested on zero-shot settings [86]. Furthermore, they are generating interest among the incremental learning community [159] due to their intersing properties as learners. As a future work, it would be interesting to study the behavior of these multi-modal models either for learning novel classes from very limited annotated images (e.g., few-shot and zero-label semantic segmentation), or to exploit their properties to extract semantic segmentation pseudo-supervision from image-level labels, thus reducing the burden of collecting and annotating a large dataset.

Incremental Learning Beyond Semantic Segmentation. In this work we deeply examined incremental learning in the semantic segmentation task, going beyond it only in Chapter 3 considering object detection and instance segmentation. However, while being fundamental for multiple application, semantic segmentation is only a part of the whole segmentation task. Recently, mask-based transformer-based architectures [31, 30, 191] demonstrated outstanding performance on both semantic, instace, and panoptic segmentation tasks, achieving state-of-the-art results on every task without changing either the architecture or the loss functions. As a next step, we aim to investigate incremental learning with these architectures, addressing seamlessy a broader range of applications.

References

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

[2] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4981–4990, 2018.

[3] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2209–2218, 2019.

[4] Zeynep Akata, Honglak Lee, and Bernt Schiele. Zero-shot learning with structured embeddings. 09 2014.

[5] Zeynep Akata, Florent Perronnin, Zaïd Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *CoRR*, abs/1503.08677, 2015. URL http://arxiv.org/abs/1503.08677.

[6] Emanuele Alberti, Antonio Tavera, Carlo Masone, and Barbara Caputo. Idda: a large-scale multi-domain dataset for autonomous driving. *IEEE Robotics and Automation Letters*, 5(4):5526–5533, 2020.

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

[8] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4253–4262, 2020.

[9] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *CoRR*, abs/1908.02983, 2019. URL http://arxiv.org/abs/1908.02983.

[10] Sarkhan Badirli, Zeynep Akata, and Murat Dundar. Bayesian zero-shot learning. *CoRR*, abs/1907.09624, 2019. URL http://arxiv.org/abs/1907.09624.

[11] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pages 549–565. Springer, 2016.

[12] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059, 2019.

[13] A Boguszewski, D Batorski, N Ziemba-Jankowska, A Zambrzycka, and T Dziedzic. Landcover. ai: Dataset for automatic mapping of buildings. *Woodlands and Water from Aerial Imagery*, 2005:02264, 2020.

[14] Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. Generating visual representations for zero-shot classification. *CoRR*, abs/1708.06975, 2017. URL http://arxiv.org/abs/1708.06975.

[15] Maxime Bucher, Tuan-Hung Vu, Mathieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. In *NeurIPS*, 2019.

[16] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Region-based semantic segmentation with end-to-end training. In *European Conference on Computer Vision*, pages 381–397. Springer, 2016.

[17] Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. *CoRR*, abs/1612.03716, 2016. URL http://arxiv.org/abs/1612.03716.

[18] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.

[19] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

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

[21] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9233–9242, 2020.

[22] Yu-Ting Chang, Qiaosong Wang, Wei-Chih Hung, Robinson Piramuthu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Weakly-supervised semantic segmentation via sub-category exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8991–9000, 2020.

[23] Arslan Chaudhry, Puneet K Dokania, and Philip HS Torr. Discovering class-specific pixels for weakly-supervised semantic segmentation. *arXiv preprint arXiv:1707.05821*, 2017.

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

[25] Li Chen, Chunyan Yu, and Lvcai Chen. A new knowledge distillation for incremental object detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2019.

[26] 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.

[27] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[28] 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.

[29] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D. Collins, Ekin D. Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens. Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation, 2020.

[30] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. *arXiv preprint arXiv:2112.01527*, 2021.

[31] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. 2021.

[32] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[33] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1635–1643, 2015.

[34] 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. *arXiv preprint arXiv:1909.08383*, 2019.

[35] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

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

[37] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, 2018.

[38] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[39] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 86–102. Springer, 2020.

[40] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Plop: Learning without forgetting for continual semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4040–4050, 2021.

[41] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. *CVPR*, 2022.

[42] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, .

[43] 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, .

[44] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

[45] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019.

[46] Dan Andrei Ganea, Bas Boom, and Ronald Poppe. Incremental few-shot instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1185–1194, 2021.

[47] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, pages 4367–4375, 2018.

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

[49] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[50] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[51] Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang. Context-aware feature generation for zero-shot semantic segmentation. *Proceedings of the 28th ACM International Conference on Multimedia*, Oct 2020. doi: 10.1145/3394171.3413593. URL http://dx.doi.org/10.1145/3394171.3413593.

[52] Akshita Gupta, Sanath Narayan, KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Ow-detr: Open-world detection transformer. *arXiv preprint arXiv:2112.01513*, 2021.

[53] Yu Hao, Yanwei Fu, Yu-Gang Jiang, and Qi Tian. An end-to-end architecture for class-incremental object detection with knowledge distillation. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2019.

[54] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*, 2011.

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

[56] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

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

[58] Qibin Hou, Peng-Tao Jiang, Yunchao Wei, and Ming-Ming Cheng. Self-erasing network for integral object attention. *arXiv preprint arXiv:1810.09821*, 2018.

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

[60] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7014–7023, 2018.

[61] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 603–612, 2019.

[62] Dana E Ilea and Paul F Whelan. Image segmentation based on the integration of colour–texture descriptors—a review. *Pattern Recognition*, 44(10-11):2479–2501, 2011.

[63] Sergey Ioffe. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. In *NeurIPS*, pages 1945–1953, 2017.

[64] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[65] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. *CoRR*, abs/1904.04717, 2019. URL http://arxiv.org/abs/1904.04717.

[66] Jitesh Jain, Anukriti Singh, Nikita Orlov, Zilong Huang, Jiachen Li, Steven Walton, and Humphrey Shi. Semask: Semantically masked transformers for semantic segmentation. *arXiv preprint arXiv:2112.12782*, 2021.

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

[68] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 876–885, 2017.

[69] 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.

[70] Joseph Kj, Jathushan Rajasegaran, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Incremental object detection via meta-learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2021. ISSN 1939-3539. doi: 10.1109/tpami.2021.3124133. URL http://dx.doi.org/10.1109/TPAMI.2021.3124133.

[71] Marvin Klingner, Andreas Bär, Philipp Donn, and Tim Fingscheidt. Class-incremental learning for semantic segmentation re-using neither old data nor old labels. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2020.

[72] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European conference on computer vision*, pages 695–711. Springer, 2016.

[73] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *CoRR*, abs/1610.02242, 2016. URL http://arxiv.org/abs/1610.02242.

[74] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014. doi: 10.1109/TPAMI.2013.140.

[75] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.

[76] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5267–5276, 2019.

[77] Seungho Lee, Minhyun Lee, Jongwuk Lee, and Hyunjung Shim. Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5495–5505, 2021.

[78] Dawei Li, Serafettin Tasci, Shalini Ghosh, Jingwen Zhu, Junting Zhang, and Larry Heck. Rilod: Near real-time incremental learning for object detection at the edge, 2019.

[79] Yanan Li and Donghui Wang. Zero-shot learning with generative latent prototype model. *CoRR*, abs/1705.09474, 2017. URL http://arxiv.org/abs/1705.09474.

[80] Yi Li, Zhanghui Kuang, Liyang Liu, Yimin Chen, and Wayne Zhang. Pseudo-mask matters in weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6964–6973, 2021.

[81] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE T-PAMI*, 40(12): 2935–2947, 2017.

[82] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3159–3167, 2016.

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

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

[85] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[86] Yuqi Lin, Minghao Chen, Wenxiao Wang, Boxi Wu, Ke Li, Binbin Lin, Haifeng Liu, and Xiaofei He. Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation. *arXiv preprint arXiv:2212.09506*, 2022.

[87] Haiyang Liu, Yichen Wang, Jiayi Zhao, Guowu Yang, and Fengmao Lv. Learning unbiased zero-shot semantic segmentation networks via transductive transfer, 2020.

[88] Liyang Liu, Zhanghui Kuang, Yimin Chen, Jing-Hao Xue, Wenming Yang, and Wayne Zhang. Incdet: In defense of elastic weight consolidation for incremental object detection. *IEEE Transactions on Neural Networks and Learning Systems*, 32 (6):2306–2319, 2021. doi: 10.1109/TNNLS.2020.3002583.

[89] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[90] Xialei Liu, Hao Yang, Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Multi-task incremental learning for object detection. *arXiv preprint arXiv:2002.05347*, 2020.

[91] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 12245–12254, 2020.

[92] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 2544–2553, 2021.

[93] Yu Liu, Sarah Parisot, Gregory Slabaugh, Xu Jia, Ales Leonardis, and Tinne Tuytelaars. More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, pages 699–716. Springer, 2020.

[94] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[95] Vincenzo Lomonaco, Davide Maltoni, and Lorenzo Pellegrini. Rehearsal-free continual learning over small non-iid batches. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 989–998. IEEE Computer Society, 2020.

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

[97] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pages 6448–6458. PMLR, 2020.

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

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

[100] Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. Adagraph: Unifying predictive and continuous domain adaptation through graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6568–6577, 2019.

[101] Andrea Maracani, Umberto Michieli, Marco Toldo, and Pietro Zanuttigh. Recall: Replay-based continual learning in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7026–7035, 2021.

[102] 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.

[103] Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. In *ICCV-WS*, pages 0–0, 2019.

[104] Umberto Michieli and Pietro Zanuttigh. Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1114–1124, 2021.

[105] Tomas Mikolov, Ilya Sutskever, Kai Chen, G.s Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 10 2013.

[106] Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. Semi-supervised semantic segmentation with high- and low-level consistency. *CoRR*, abs/1908.05724, 2019. URL http://arxiv.org/abs/1908.05724.

[107] Takeru Miyato, Shin ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning, 2018.

[108] Tanmoy Mukherjee and Timothy Hospedales. Gaussian visual-linguistic embedding for zero-shot recognition. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 912–918, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1089. URL https://www.aclweb.org/anthology/D16-1089.

[109] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *International Conference on Computer Vision (ICCV)*, 2017. URL https://www.mapillary.com/dataset/vistas.

[110] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[111] Seong Joon Oh, Rodrigo Benenson, Anna Khoreva, Zeynep Akata, Mario Fritz, and Bernt Schiele. Exploiting saliency for object segmentation from image level labels. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 5038–5047. IEEE, 2017.

[112] 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.

[113] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training, 2020.

[114] 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.

[115] 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.

[116] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot learning with semantic output codes. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22, pages 1410–1418. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper/2009/file/1543843a4723ed2ab08e18053ae6dc5b-Paper.pdf.

[117] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1742–1750, 2015.

[118] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1796–1804, 2015.

[119] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

[120] Can Peng, Kun Zhao, and Brian C. Lovell. Faster ilod: Incremental learning for object detectors based on faster rcnn, 2020.

[121] Can Peng, Kun Zhao, Sam Maksoud, Tianren Wang, and Brian C. Lovell. Diode: Dilatable incremental object detection, 2021.

[122] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M Hospedales, and Tao Xiang. Incremental few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13846–13855, 2020.

[123] Hieu Pham, Qizhe Xie, Zihang Dai, and Quoc V. Le. Meta pseudo labels, 2020.

[124] Pedro O Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1713–1721, 2015.

[125] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *CVPR*, pages 5822–5830, 2018.

[126] Rui Qian, Yunchao Wei, Honghui Shi, Jiachen Li, Jiaying Liu, and Thomas Huang. Weakly supervised scene parsing with point-based distance metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8843–8850, 2019.

[127] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[128] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alexei A Efros, and Sergey Levine. Few-shot segmentation propagation with guided networks. In *ICLR-W*, 2018.

[129] Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. Semi-supervised learning with ladder network. *CoRR*, abs/1507.02672, 2015. URL http://arxiv.org/abs/1507.02672.

[130] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[131] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30, 2017.

[132] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017.

[133] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[134] Mengye Ren, Renjie Liao, Ethan Fetaya, and Richard S. Zemel. Incremental few-shot learning with attention attractor networks. 2019.

[135] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[136] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *European Conference on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016.

[137] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

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

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

[140] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2019.

[141] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

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

[143] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *BMVC*, 2017.

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

[145] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *ICCV*, 2017.

[146] Mennatullah Siam, Boris Oreshkin, and Martin Jagersand. Adaptive masked proxies for few-shot segmentation. *ICCV*, 2019.

[147] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, pages 4077–4087, 2017.

[148] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

[149] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, pages 7262–7272, 2021.

[150] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *European conference on computer vision*. Springer, 2020.

[151] Qianru Sun, Xinzhe Li, Yaoyao Liu, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. *CoRR*, abs/1906.00562, 2019. URL http://arxiv.org/abs/1906.00562.

[152] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, pages 1199–1208, 2018.

[153] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[154] Meng Tang, Abdelaziz Djelouah, Federico Perazzi, Yuri Boykov, and Christopher Schroers. Normalized cut loss for weakly-supervised cnn segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1818–1827, 2018.

[155] Meng Tang, Federico Perazzi, Abdelaziz Djelouah, Ismail Ben Ayed, Christopher Schroers, and Yuri Boykov. On regularized losses for weakly-supervised cnn segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 507–522, 2018.

[156] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *CVPR*, pages 12183–12192, 2020.

[157] Antti Tarvainen and Harri Valpola. Weight-averaged consistency targets improve semi-supervised deep learning results. *CoRR*, abs/1703.01780, 2017. URL http://arxiv.org/abs/1703.01780.

[158] 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.

[159] Vishal Thengane, Salman Khan, Munawar Hayat, and Fahad Khan. Clip model is an efficient continual learner. *arXiv preprint arXiv:2210.03114*, 2022.

[160] Neil C Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F Manso. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*, 2020.

[161] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *ECCV*. Springer, 2020.

[162] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[163] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[164] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 32–42, 2021.

[165] Gido M van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, pages 1–13, 2022.

[166] Vinay Kumar Verma and Piyush Rai. A simple exponential family framework for zero-shot learning. *CoRR*, abs/1707.08040, 2017. URL http://arxiv.org/abs/1707.08040.

[167] Paul Vernaza and Manmohan Chandraker. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7158–7166, 2017.

[168] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *NeurIPS*, 29:3630–3638, 2016.

[169] Riccardo Volpi, Diane Larlus, and Grégory Rogez. Continual adaptation of visual representations via domain randomization and meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4443–4453, 2021.

[170] Riccardo Volpi, Pau De Jorge, Diane Larlus, and Gabriela Csurka. On the road to online adaptation for semantic image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19184–19195, 2022.

[171] Bin Wang, Guojun Qi, Sheng Tang, Tianzhu Zhang, Yunchao Wei, Linghui Li, and Yongdong Zhang. Boundary perception guidance: a scribble-supervised semantic segmentation approach. In *IJCAI International Joint Conference on Artificial Intelligence*, 2019.

[172] Junjue Wang, Zhuo Zheng, Ailong Ma, Xiaoyan Lu, and Yanfei Zhong. Loveda: A remote sensing land-cover dataset for domain adaptive semantic segmentation. *arXiv preprint arXiv:2110.08733*, 2021.

[173] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *ICCV*, pages 9197–9206, 2019.

[174] Kaixin Wang, JunHao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. *CoRR*, abs/1908.06391, 2019. URL http://arxiv.org/abs/1908.06391.

[175] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12275–12284, 2020.

[176] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. *ECCV*, 2022.

[177] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *CVPR*, pages 139–149, 2022.

[178] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1568–1576, 2017.

[179] Daniel Weinland, Remi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer vision and image understanding*, 115(2):224–241, 2011.

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

[181] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90:119–133, 2019.

[182] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh N. Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. *CoRR*, abs/1603.08895, 2016. URL http://arxiv.org/abs/1603.08895.

[183] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly. *CoRR*, abs/1707.00600, 2017. URL http://arxiv.org/abs/1707.00600.

[184] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. *CoRR*, abs/1712.00981, 2017. URL http://arxiv.org/abs/1712.00981.

[185] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero- and few-label semantic segmentation. In *CVPR*, June 2019.

[186] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *NeurIPS*, 34, 2021.

[187] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2021.

[188] Dongbao Yang, Yu Zhou, and Weiping Wang. Multi-view correlation distillation for incremental object detection, 2021.

[189] Yazhou Yao, Tao Chen, Guo-Sen Xie, Chuanyi Zhang, Fumin Shen, Qi Wu, Zhenmin Tang, and Jian Zhang. Non-salient region object mining for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2623–2632, 2021.

[190] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *ICLR*, 2016.

[191] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means mask transformer. In *European Conference on Computer Vision*, pages 288–307. Springer, 2022.

[192] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018.

[193] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. *ECCV*, 2020.

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

[195] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *CVPR*, pages 5217–5226, 2019.

[196] Lei Zhang, Peng Wang, Lingqiao Liu, Chunhua Shen, Wei Wei, Yanning Zhang, and Anton van den Hengel. Towards effective deep embedding for zero-shot learning. *CoRR*, abs/1808.10075, 2018. URL http://arxiv.org/abs/1808.10075.

[197] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas S Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *IEEE Transactions on Cybernetics*, 2020.

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

[199] Ziming Zhang and Venkatesh Saligrama. Classifying unseen instances by learning class-independent similarity functions. 11 2015.

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

[201] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

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

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

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

[205] Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. Pseudoseg: Designing pseudo labels for semantic segmentation, 2020.