Politecnico di Torino

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

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

Recognizing New Visual Categories in Real Open World Environments

By

Dario Fontanel
******

Supervisor(s):
Prof. Barbara Caputo

Doctoral Examination Committee:

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.

Dario Fontanel
2023

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

Acknowledgements

In this segment, I would like to express my gratitude to all those who accompanied me on this incredible journey. Heartfelt thanks to my Ph.D. advisor, Prof. Barbara Caputo, who took me under her wing when I was a master student that didn't know what to do with his life. Thank you for supporting me throughout these years, for every piece of advice you gave me, and more importantly, thank you for teaching me that believing in yourself will get you far. I would like to express my gratitude also to Prof. Pietro Zanuttigh and and Prof. Luca Iocchi for having taken the time to accurately read this thesis. Thank you for your positive and valuable feedback.

This journey would not have been the same without good companions along the way. Huge thanks to Fabio for being an invaluable friend before than a co-author, a teammate, and a role model. Thank you for everything you taught me, not only as a researcher. Thanks also to Antonio, with whom I have shared countless efforts while working together. Thanks to Chiara for always brightening my day with her joyful attitude, and for genuinely caring about me. Thanks also to Massi, who has always watched over me, and taught me that true hard work will always beat talent. Thanks to Niccolò, Matteo, Antonino and Mauro for helping me in becoming a better supervisor. Thanks to Silvia, Mirco, Francesco, Debora, Antonio and every member of the VANDAL lab for sharing with me unforgettable memories about the lab and the conferences.

This achievement would have been possible without family and friends supporting me. I cannot express in words how thankful I am to my parents, who have always stood by my side. I hope that the culmination of this journey can bring them even a fraction of the joy they have given me. Thanks to my sister, Liviana, and to my brother, Davide, for their constant encouragement. I know we are scattered all over the world lately, but every time we see each other it feels like no distance has separated us. Lastly, I am immensely grateful to my closest friends, without whom

my life would not be the same. Thanks to Valerio, who has always had my back, and has constantly inspired me with his ambition and noble values. Thanks to Aronne, a treasured source of advice, travels planning, laughs, and amazing memories. Finally, thanks to Simone, who, despite the physical distance, has always stayed close to me, helping me when I needed the most.

Abstract

Deep learning is the dominant approach in modern computer vision. However, its success mainly hinges on the availability of large scale annotated datasets for training, and capturing the infinite semantic diversity of the real world into one or more training sets is, unfortunately, unfeasible. Consequently, deep neural networks are forced to limit their understanding of the world to the restricted knowledge available during the training phase. In this thesis, we argue that, to develop deep neural networks capable of operating in the real world, it is vital to empower them with the capabilities of i) detecting previously unseen concepts and ii) incrementally integrating them in subsequent learning stages. In the first part of this thesis, we address the aforementioned challenges separately. We first address the anomaly segmentation problem, which involves identifying for each pixel of an image whether it belongs to a previously unseen category, *i.e*. an anomaly. We propose to segment anomalies using class-specific prototypes extracted from a cosine classifier, and to determine pixels to be anomalous when the highest matching score between a pixel and the set of known prototypes is below a certain threshold. We then address the challenges of incremental learning, which involves incrementally updating existing models as new categories become available. Despite advancements in the field, state-of-the-art semantic segmentation strategies still require supervision at pixel-level on new classes, which is often costly and time-consuming to acquire. In this thesis we present a new perspective to the field, showing how to incrementally extend the knowledge of a pre-trained segmentation model using only cheap image-level labels, which provide information only on the presence of a certain class but not on its shape or location. We demonstrate that directly applying existing weakly-supervised segmentation strategies to the traditional incremental segmentation ones is sub-optimal, and we propose to use a localizer module to produce pseudo-labels and a distillation-based loss to prevent forgetting previously learned classes. In the second part of this thesis, we address the open world recognition (OWR) setting to tackle the

two challenges simultaneously. Differently from prior works, we demonstrate that learning a separate rejection threshold for each class is crucial to reduce the number of samples wrongly identified as never-seen-before ones. To achieve this, we shape the representation space to be semantically consistent through a global-to-local clustering approach, that enforces samples to be closer to the centroid of the respective class, while pushing away samples from other classes. The training sets, however, impose not only semantic limitations on agents, but also environmental ones, due to the inherent bias towards specific acquisition conditions that do not necessarily represent the high variability of the real world. Therefore, in the final part of the thesis, we investigate the impact of different training and test distributions (domain-shifts) on OWR frameworks. We introduce the first benchmark to assess OWR methods under domain-shifts, and we show that existing OWR strategies significantly suffer from performance degradation when the train and test distributions differ. We demonstrate that coupling OWR methods with domain generalization algorithms mitigates this degradation, but their simple integration is not sufficient to identify new and unknown categories in unfamiliar domains. We then highlight open challenges and future research directions, that serve as foundations towards developing agents capable of reliably operating in real open world environments.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A long-standing goal of the current artificial intelligence and robotics revolution is to create agents capable of autonomously operating in the real world. In order to achieve this goal, it is essential for agents to understand their surrounding environment, which is made possible through powerful sensors that act as windows into the real world. Visual cameras are among the most powerful, accessible and informative sensors, and have become extremely valuable in applications that require visual capabilities, such as self-driving vehicles, warehouse robots, indoor vacuum robots, etc. All of these applications, in order to operate in a real world setting, require a deep understanding of the appearance, characteristics, and functions of their surroundings. Robot vision systems have made remarkable progress in recent years, thanks to the use of deep learning architectures, specifically Convolutional Neural Networks (CNNs), which achieved outstanding results in a variety of computer vision applications, ranging from object classification and detection, to semantic segmentation, scene understanding, action recognition, object tracking, and more.

Despite their effectiveness, traditional deep neural networks have one major limitation preventing them to operate in a real world environment. Relying on the *closed world assumption (CWA)*, which implies that all the classes a model will ever need to recognize are present in the training set(s), deep neural networks confine their understanding of the real world only to the knowledge available during the training phase. Obviously, this perspective is extremely limiting, as the real world contains an endless set of potential input conditions (e.g. various illumination, environments), and because models are likely to encounter previously unseen categories after being

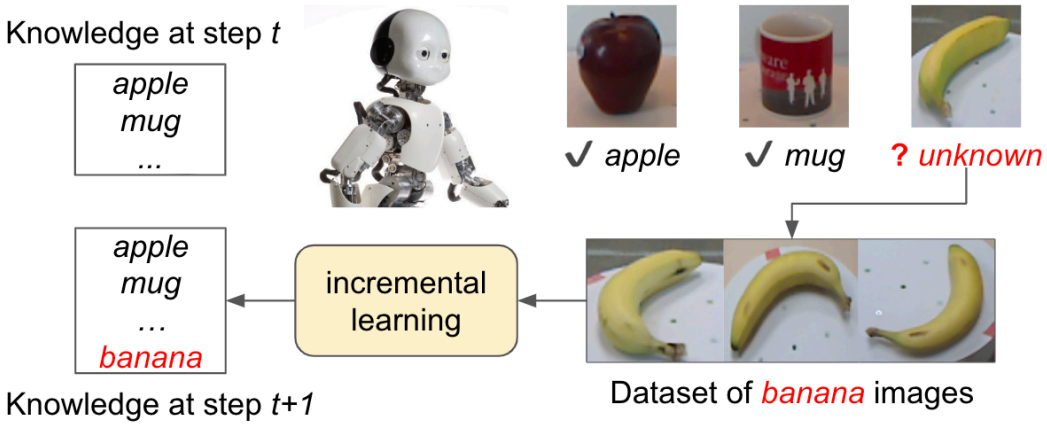


Fig. 1.1 In the context of an open world scenario, it is essential for an agent to possess the ability to accurately classify known objects (such as *apple* and *mug*), while also being capable of detecting novel semantic concepts (such as *banana*). When a novel concept is identified, the agent should learn the new category and update its knowledge base accordingly.

deployed, it is crucial to empower them with the ability to detect these novel classes. However, identifying unknowns is not the only challenge that deep learning models encounter in real world environments. After deployment, it is likely that these models will require to be updated to recognize those new semantic categories, while also avoiding forgetting already learned concepts. In this thesis, we show how to break the CWA, getting one step closer towards the development visual systems capable of operating in the *open world*.

To clarify our objective, let us examine the example reported in Fig. 1.1. The knowledge base of our robot consists of a limited set of categories (e.g., apple, mug), and the robot is capable of detecting and classifying only the objects belonging to them. However, it is highly probable that the robot will encounter a new object (e.g., banana) at some stage during its lifespan. In such a situation, we need the robot to detect the novel object and categorize it as unknown. Additionally, once a set of images for that object is acquired, the robot needs to update its knowledge base and learn how to detect and classify the new object.

To achieve this goal, a robot vision system must possess two essential capabilities: (i) it must have the capacity to recognize concepts it has already encountered, while also detecting previously unseen ones, and (ii) it must be capable of expanding its knowledge base with novel categories without erasing the previously learned ones, and without having access to previous training datasets.

In the initial part of the thesis, we address the first issue, investigating the challenges of identifying unknown objects at pixel level. This scenario is referred to as *anomaly segmentation (AS)*, which entails segmenting an image and determining the class to which each pixel belongs if known, otherwise recognizing the pixel as *anomalous*. We propose a novel approach based on prototype learning, which achieves the new state-of-the-art in identifying anomalous objects at the pixel level, while maintaining accuracy in recognizing previously seen classes. We note that, in this context, we define anomalies only as pixels belonging to never-seen-before categories, which differs from the definition of anomalies in industrial contexts, where it typically refers to defects or unexpected behaviors.

The second part of this thesis focuses on the complementary scenario, referred to as *incremental learning (IL)*, which involves incrementally updating existing models as new classes become available. While state-of-the-art segmentation models still require full pixel-wise labels, that are costly and time-consuming to acquire, we introduce a new perspective to the field by focusing on the more realistic challenge of extending the knowledge of a pre-trained segmentation model using only cheap image-level labels. These image-level labels provide information only about the presence or absence of a particular class, without providing any cues on their location or appearance. We present a novel approach that employs a *localizer* module for generating pseudo-labels, and a distillation-based loss to avoid forgetting previously learned categories, resulting in the new state-of-the-art.

In the final section of the thesis, we address the two scenarios simultaneously in the *open world recognition (OWR)* setting. Firstly, we present our method, B-DOC, which employs two complementary clustering losses to learn semantically consistent clusters in the representation space. This allows us to identify unknown samples by learning distance-based class-specific rejection thresholds. We then investigate how the differences between training and test domains (*i.e.*, the domain-shift problem) affect open world recognition algorithms by introducing the first benchmark in the field. Our results show that OWR algorithms perform significantly worse when evaluated on different domains, and coupling them with single-source domain generalization strategies can only reduce but not solve the issue.

1.1 Contributions

Focusing on visual recognition, this thesis contributes towards developing deep learning architectures able to operate in the real world, empowering models with the capabilities of identifying previously unseen categories, and learning them afterwards. The main contributions of this work can be divided into three parts. Firstly, we present a prototypes-based approach to detect previously unseen objects at pixel level, while maintaining the ability of the model to detect and recognize already learned classes. Second, we present a method to incrementally extend the knowledge of a segmentation module over time without the need for full pixel-wise traditional labels, using cheaper and easier to obtain image-level labels instead. Finally, we address both challenges in one single fashion, introducing a clustering paradigm able to cluster samples in the feature space and learn class-specific rejection thresholds to distinguish between known and unknown categories. Additionally, we investigate the impact of domain-shift on open world recognition algorithms, and identify potential future directions.

Specifically, we present:

a prototype-based approach to segment at pixel levels known and unknown categories [16]. The intuition is that learning generalized but distinctive representations of each class allows the model to identify anomalies as pixels that do not match any of these representations. We obtain the prototypes using a cosine classifier that encodes the corresponding average pixel features for each known class, and we consider a pixel to be anomalous only if the highest matching score with the known classes is below a certain threshold.

a novel framework capable of segmenting previously unseen classes using only cheap image-level labels [17]. Unlike previous methods which generate pseudo-labels offline, we train a *localizer* module using only image-level labels, which enables us to obtain online pseudo-supervision and update the model incrementally. Coupling this module with a distillation-inspired loss, we are able to continuously extend the knowledge of the model to segment new classes, while avoiding forgetting already learned ones.

a global-to-local features clustering approach that allows us to learn class-specific rejection thresholds [13]. We introduce a global clustering loss term that enforces the model to map samples closer to the centroid of their class, and a local

clustering loss term that brings samples of the same class closer and pushes away neighbors from other categories. Moreover, differently from previous works, we learn class-specific rejection thresholds by introducing a novel loss formulation, rather than using a single global threshold.

a case study on the effectiveness of open world algorithms under domain-shifts [18]. We present the first benchmark to fairly evaluate open world algorithms with and without domain-shifts. Our findings show that existing methods experience significant performance degradation when trained and tested on different distributions. Additionally, we show that simply integrating domain generalization strategies into open world algorithms only partially alleviates this degradation. Our results point towards open challenges and opportunities for future research in developing robust visual systems for robots that can operate under such challenging yet realistic conditions.

1.2 Outline

Chapter 2 introduces the closed world assumption (CWA) (Section 2.1), being one of the major limitations for deep models to operate in the real world, and proposes a strategy to deal with it. Section 2.2 presents a literature review that serves as starting point for our work. It gives the foundation for semantic segmentation, out-of-distribution detection, and anomaly segmentation. Section 2.3.1 presents the anomaly segmentation more in detail, providing a precise mathematical formulation, and analyzing the drawbacks of the popular approach MSP, that uses the highest probability assigned to any of the known classes to infer the anomaly score for a pixel. In Section 2.3.2 we introduce our method **PAnS**, that leverages a cosine classifier to learn a class-specific prototype that allows us to compute anomaly scores from the classification scores directly, overcoming the softmax function limitations of MSP. The experimental setting and results in Section 2.3.3 support the effectiveness of our method.

Chapter 3 leads us to incremental learning (IL), which is a crucial step in developing models able to work in realistic environments. Section 3.1 presents IL challenges, while Section 3.2 provides an extensive review of related work. In Section 3.3 we introduce a more realistic setting with respect to traditional scenarios, in which we aim at extending the knowledge of a pre-trained deep model using only cheap image-level labels. We start with a mathematical formulation of the problem setting (Section 3.3) and we present our method **WILSON**, that couples the segmentation model with a localizer module, and uses image-level labels on new categories to generate pseudo-labels to train the segmentation backbone in a single fashion. We detail the components of WILSON in Section 3.3.2 and 3.3.3, and we present our qualitative and quantitative results in Section 3.3.4, which show how WILSON is able to outperform weakly-supervised semantic segmentation methods, and achieve similar results compared to standard fully supervised IL approaches.

Chapter 4 introduces the open world recognition (OWR) setting that ultimately aims at breaking the CWA, empowering models to identify the presence of unknown categories as well as to learn new ones as they become available for training. We introduce the problem in Section 4.1, and we provide a review of OWR strategies in Section 4.2. In Section 4.3 we present our method **B-DOC**, which introduces a global-to-clustering loss training objective, and trainable class-specific rejection thresholds to distinguish between known and unknown categories. Section 4.3.3 presents our

experimental results and findings. In Section 4.4 we take a step further, and investigate the effects that domain-shifts have on OWR frameworks. Section 4.4.1 presents the first benchmark to assess OWR algorithms under shifting visual domains, and Section 4.4.2 presents our findings. In particular, we asses how domain-shift affects OWR methods in Section 4.4.3, and we investigate whether domain generalization approaches can solve the degradation in performance in Section 4.4.4.

The thesis concludes by summarizing our findings, open challenges, and possible future direction of research in **Chapter 5**.

1.3 Publications

The following list provides an overview of the author's publications in chronological order; note that some of the published papers (marked with *) are excluded from this thesis.

- D. Fontanel, F. Cermelli, M. Mancini, S. Rota Bulò, R. Ricci, B. Caputo
  Boosting Deep Open World Recognition by Clustering
  IEEE Robotics and Automation Letters 2020, vol. 5, n. 4, pp. 5985-5992. Presented at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2020.

- D. Fontanel, F. Cermelli, M. Mancini, B. Caputo
  On the challenges of open world recognition under shifting visual domains
  IEEE Robotics and Automation Letters 2020, vol. 6, n. 2, pp. 604-611. Presented at IEEE International Conference on Robotics and Automation (ICRA) 2021.

- D. Fontanel, F. Cermelli, M. Mancini, B. Caputo
  Detecting anomalies in semantic segmentation with prototypes
  IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2021.

- F. Cermelli, D. Fontanel, A. Tavera, M. Ciccone, B. Caputo
  Incremental learning in semantic segmentation from image labels
  IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022.

- * F. Cermelli, A. Geraci, D. Fontanel, B. Caputo
  Modeling missing annotations for incremental learning in object detection
  IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022.

- * D. Fontanel, F. Cermelli, A. Geraci, M. Musarra, M. Tarantino, B. Caputo
  Relaxing the Forget Constraints in Open World Recognition
  International Conference on Image Analysis and Processing (ICIAP) 2022.

- * D. Fontanel, D. Higham, B. Quantin Arthur Vallade
  On the Importance of Spatio-Temporal Learning for Video Quality Assessment

IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) 2023.

Chapter 2

Recognizing unseen semantic concepts

Traditional semantic segmentation methods can recognise during testing only the specific set of classes they have seen in the training phase, which is a major limitation for autonomous systems operating in realistic environments. Unexpected, unknown objects will inevitably appear during testing, and the inability to identify such anomalies might lead autonomous agents equipped with such segmentation modules to incorrect or even dangerous behaviour. The majority of state-of-the-art anomaly segmentation approaches relies on generative models, which are expensive and prone to consider generated artifacts as false anomalies. In this chapter we take a different direction. In particular, we start by presenting an overview of the problem in Section 2.1, followed by a review of the related literature Section 2.2. In Section 2.3, we present ***PAnS*** *(**P**rototopycal **An**omaly **S**egmentation) which uses prototype learning to segment anomalies. We extract prototypes from the training data by means of lightweight cosine similarity-based classifier. Experiments (Section 2.3.3) on the popular StreetHazards benchmark confirm the effectiveness of our method.*

2.1 Problem statement

As highlighted in Chapter 1, for machines operating in the real world environments, it is essential to be able to identify which objects are present in their surroundings, and where. To achieve this goal, several works have focused on the semantic segmentation task [19, 20], in which the objective is to assign a semantic label to each pixel in an image. Semantic segmentation models, however, are inherently bounded to recognise only the classes they see annotated during training (closed world assumption). Regardless of how vast their training database may be, it is clearly not possible to anticipate and capture every potential semantic class a system may encounter. Ideally, a segmentation model would need to be able to recognize when a pixel belongs to one of its known classes, or when it belongs to an unseen category that was not included in the training set. This capability is especially important in cases where pixels of unknown categories could potentially become a threat to the machine (or the human) using the semantic segmentation module. As an example, consider the scenario depicted in Fig. 2.1. Because no semantic segmentation dataset provides labeled pixels for the *helicopter* class, the model has no chance of avoiding a fatal collision, unless it detects that there is something unexpected in the image.

In this chapter, we focus on the problem known as anomaly segmentation (AS) [21, 22], which involves recognizing whether a pixel in an image belongs to a category that the model was not trained on, *i.e.* an *anomaly*. Previous approaches to this problem have either imposed a threshold on the predicted probabilities for each pixel [23] or employed generative methods to compare input images (or features) to their reconstructed counterparts [24, 22]. However, both of these strategies have their limitations. The first approach ignores the fact that the softmax function blurs the model's confidence regarding the presence of a particular class. In other words, after the softmax function is applied, two classes that were predicted with high scores (logits) end up having an equal low probability. On the other hand, it is particularly difficult to generate images with high fidelity in the context of semantic segmentation because of the complexity of the content. As a result, generative approaches tend to produce artifacts not only when synthesizing pixels of unknown classes, but also when synthesizing pixels of known classes (see Fig. 2.2). These behaviours limit the effectiveness of these methods for anomaly segmentation.

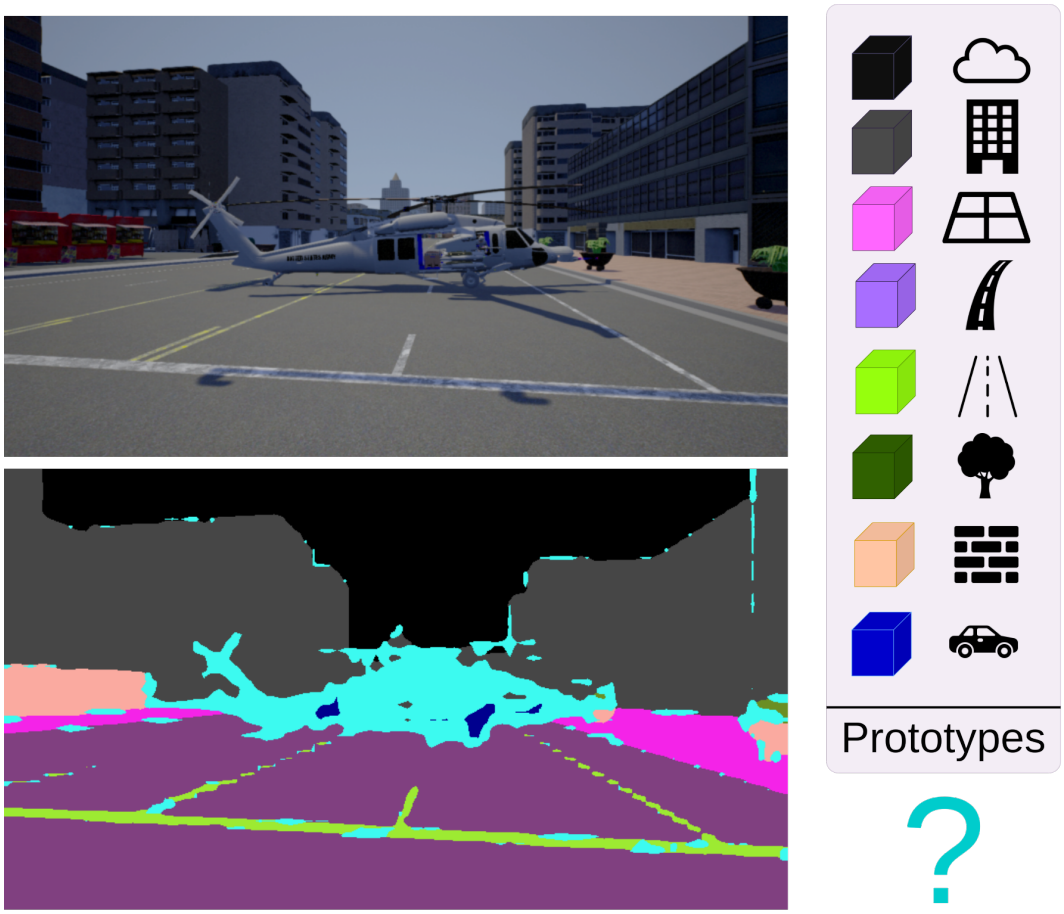


Fig. 2.1 The goal of anomaly segmentation (AS) is to segment objects that the model has not seen before. Addressing AS is critical, particularly in autonomous driving scenarios, where mistakenly identifying an anomalous object for a known one can be highly dangerous. In this chapter, we address AS via prototype learning, where anomalies (depicted in light-blue) are defined as all regions that do not match any class prototypes that the model has learned.

In this chapter, we propose to address the anomaly segmentation problem directly at the class scores level. Our intuition is that if a model learns general but discriminatory representations of each class, it will be able to detect anomalies as pixels that do not match any of the class representations. We pursue this idea by introducing class-specific prototypes and considering a pixel to be anomalous only when its highest matching score with the set of prototypes of the known classes is below a certain threshold. We extract the prototypes using a cosine classifier that embodies for each of the known classes the corresponding average pixel features. It is important to note that we avoid the normalization issues of softmax-based strategies,

since we estimate anomalies *directly* from the compatibility between the feature set of a test sample, and the prototypes of the known classes. We evaluate our model, called PAnS (**P**rototopycal **An**omaly **S**egmentation), on the popular StreetHazards benchmark [2] and demonstrate that it performs better than expensive generative approaches, while largely surpassing the previous state of the art. Additionally, a final ablation study completes our experiments.

Contributions. To summarize, the contributions presented in this chapter are three-fold. Firstly, we present a new perspective for the anomaly segmentation problem, which emphasizes the importance of class-specific scores rather than probabilities in identifying anomalous pixels. Secondly, we propose our **P**rototypical **An**omaly **S**egmentation (PAnS) method, which matches a test sample's feature vector and class-specific prototypes to compute class-specific scores, and learns the prototypes as weights of a cosine classifier. Lastly, we demonstrate through experiments on the widely adopted StreetHazards dataset that our approach surpasses the previous state-of-the-art by a margin.

2.2 Literature review

In this section we review the fundamental topics that serve as the building blocks for our work, *i.e*. semantic segmentation architectures, out-of-distribution detection and anomaly segmentation.

Semantic segmentation. Modern semantic segmentation architectures [19, 25–28] consist of fully-convolutional encoder-decoder networks [19, 29] which differ in the strategy in which contextual information is integrated into the pixel-level features. These works can be categorized into two main approaches: pyramid-based approaches [26, 27, 30, 20, 28, 25] which incorporate modules that leverage information at different scales, and attention-based approaches [31–36] which aggregate long-range spatial dependencies using attention modules at different levels. To completeness, we briefly summarize in the following semantic segmentation transformers-based architectures [37–40], which recently gained attention in the field. [37] projects image patches into a sequence of embeddings which is encoded using a transformer encoder and decoded by a mask transformer decoder to predict segmentation masks. [38] removes the positional encoding by employing a hierarchical transformer encoder which outputs multi-scale features, and a MLP decoder which combines the information coming from the different encoder layers to produce the final segmentation maps. [39] proposes to addresses segmentation as a mask classification problem. It employs a transformer and a per-pixel decoder to generate per-pixel and mask embeddings, which are then combined together to produce the segmentation output. Building upon [39], [40] introduces a transformer decoder which adopts a novel masked-attention module, and feeds the transformer decoder with one pixel-decoder feature at a time.

All of these architectures, however, have a common limitation: they necessitate a vast amount of training data, which is often both time-consuming and extremely costly to acquire. Furthermore, these models only operate in an offline scenario, meaning that it is impossible to incorporate additional knowledge after training. While recent works have attempted to address the addition of new classes [41–43, 15, 44, 14], none of these approaches account for anomaly detection.

Out-of-distribution detection aroused growing interest in the machine learning community in recent years. The authors of [23] set the baseline for out-of-distribution (OOD) detection by applying a threshold on the maximum softmax probability

(MSP) which determines whether a sample belongs to the training distribution (in-distribution) or not (out-of-distribution). Besides being simple and effective, MSP is not optimal to detect anomalies for mainly two reasons. To begin with, the model might produce high probabilities even when the predictions are incorrect [45]. Moreover, in cases where the predictions are correct but with low probability values, the model might misinterpret them as OOD samples. Other methods have investigated OOD detection challenges and proposed alternative solutions. Early works [46, 47] used Monte Carlo Dropout (MC-Dropout) to compute the uncertainty of the model, by performing multiple forward passes of the same input image randomly selecting each time a different dropout probability. [48] proposes identifying class-wise highly similar training samples and removing the sparse samples that might be outliers. At test time, the model adopts a modified nearest-neighbor classifier, which computes the prediction based on the distances between class sets. [45] takes a different approach, and uses in-distribution data to train an additional neural network with the objective of emitting high confidence values when the original model's predictions are correct. At test time, the additional module is used to detect if the predictions of the main network are reliable or not. ODIN [49] improved the baseline proposed by [23] introducing a perturbation over the features before the classification stage, and a scaling factor into the subsequent softmax operation. Similarly, [50] scales the softmax probabilities by a temperature factor, and for each sample it computes the energy which is higher for known samples rather than unobserved ones. Both of these hyperparameters were computed on an out-of-distribution validation set. Differently, [51] uses Mahalanobis distance to learn a confidence score, while [52] introduces an entropy-based classifier to detect OOD classes.

It is worth noting that our method (depicted in Section 2.3.2) conceptually differs from the approaches stated above mainly in two aspects. First, rather than depending on a temperature scaling factor for the softmax function, we compute the class scores directly through prototype-matching. Second, whereas [53, 51, 52, 24] require an external out-of-distribution set, we train our model using only in-distribution data.

Anomaly segmentation. The approaches discussed in the previous paragraph focus on determining whether an entire image is an OOD example or not. In this section, instead, we focus on the more challenging yet realistic scenario known as *anomaly segmentation* (AS) [2, 22, 54, 16, 55], in which models are asked to recognize whether each pixel of an image belongs to the in-distribution or not. OOD pixels are

referred to as *anomalies*, and AS is the task of segmenting anomalous regions from an image. Early AS approaches [55, 21] were based pixel-wise reconstruction from auto-encoders (AEs). In complex road scenes, AEs have been shown to be unable to correctly model the in-distribution in complicated road scenarios, leading to inferior performance [2] compared to less complex baselines (e.g. MSP [23]). Recent works on generative models [24, 22] achieved promising results, by measuring the discrepancies between the reconstructed version of OOD images and the original ones. The former used pix2pixHD [56] to reconstruct test images and a discrepancy network to compare them with the respective original versions, while the latter used SPADE [1] and a comparison module based on cosine similarity. One disadvantage of these approaches is that artefacts in the reconstructions might be misidentified as anomalies, as depicted in Fig. 2.2. Taking a completely different approach, the recent work of [57] proposes a hybrid approach combining the known class posterior, the dataset posterior, and an un-normalized data likelihood to estimate anomalies.

Unlike prior works, our method illustrated in Section 2.3.2 does not rely on expensive generative approaches, instead producing predictions directly from the prototypes-matching.

Fig. 2.2 Qualitative results of SPADE [1] reconstructions (top) on an image from StreetHazards dataset [2] (middle). The green box identifies the anomaly, which the model correctly does not reconstruct. The red box, on the other hand, shows one example of the artifacts that the generator produces, *i.e.* the traffic lights are not reconstructed by the model and are thus predicted as anomalies.

2.3 Anomaly Segmentation

2.3.1 Problem formulation

Let us denote as $\mathcal{X} \in \mathbb{R}^{|\mathcal{I}|}$ the image space, where $\mathcal{I}$ is the set of pixels. During training, we are given a dataset $\mathcal{T} = \{(x_k, y_k)\}_{i=k}^{N}$ where $x \in \mathcal{X}$ is an image and $y \in \mathcal{Y}$ is its corresponding ground-truth mask. As in standard segmentation, $\mathcal{Y}$ contains pixel-level annotations for a set of semantic classes $\mathcal{C}$, *i.e.* $\mathcal{Y} \in \mathcal{C}^{|\mathcal{I}|}$. Given $\mathcal{T}$, we want to learn a function f mapping an image to its corresponding anomaly score at pixel level, *i.e.* $f : \mathcal{X} \to \mathbb{R}^{|\mathcal{I}|}$. Without loss of generality, we consider f built on three components. The first is a feature extractor $\omega : \mathcal{X} \to \mathcal{Z}$ mapping images into a feature space $\mathcal{Z} \subset \mathbb{R}^{|\mathcal{I}| \times d}$, with d being the feature dimensions. The second is a scoring function $\rho : \mathcal{Z} \to \mathbb{R}^{|\mathcal{I}| \times |\mathcal{C}|}$ mapping the features in $\mathcal{Z}$ to pixel-level class scores. The third is an anomaly score function $\sigma : \mathbb{R}^{|\mathcal{I}| \times |\mathcal{C}|} \to \mathbb{R}^{|\mathcal{I}|}$, mapping the class scores to the final anomaly ones.

In the following, we will examine how previous approaches have instantiated the σ function, which is responsible for producing the final anomaly scores.

Maximum Softmax Probability (MSP) [23] is one of the most widely-used approaches for anomaly segmentation. The idea behind MSP is that the anomaly score of a pixel should be determined depending on the highest probability assigned to any of the known classes. Given a test image x and the corresponding pixel-level class scores $s = \rho(\omega(x))$, MSP computes the anomaly score for pixel i, denoted as $\sigma_i(s)$, as follows:

$$\sigma_i(s) = 1 - \max_{c \in \mathcal{C}} \frac{e^{s_i^c}}{\sum_{k \in \mathcal{C}} e^{s_i^k}} \tag{2.1}$$

where $s_i^k = \rho_k^i(z)$ is the score for class k in pixel i. It is worth noting that MSP defines the anomaly scores as the inverse of the maximum probability assigned to any of the known classes in $\mathcal{C}$, and that the probabilities are computed using the softmax function.

Also known as normalized exponential function, the softmax function, is the most commonly used function to convert the logits produced by a model into class probabilities. For each score, it takes its exponents value and divides it by the sum of the exponents of all the scores so that the output vector adds up to 1, becoming a probability score. Despite its effectiveness, we argue that utilising softmax probabil-

ity values to estimate anomaly scores is not the best approach. In fact, when using the softmax function, the confidence of the model on each pixel may get smoothed, sometimes flattened, leading pixels with high predicted initial scores to be considered uncertain (and thus anomalous) after the softmax normalization.

As a toy example, let us suppose we have two different classes and a model trained to classify pixels between them. Given a test pixel, if the model produces a very high score for one class and a very low score for the other one, the probabilities computed by the softmax function will correctly have a low entropy, with a high probability for the class with the highest score and a low probability for the other. Therefore, the model will correctly classify the pixel as belonging to the class with the higher score, and consider it to be not anomalous. On the other hand, if the scores for both classes are high but closed in values, the probabilities produced by the softmax function will have high entropy after the softmax normalization. In this case, this high entropy indicates that the model is uncertain about the semantics of the pixel, but the high initial scores may suggest that the model *is not uncertain* that the pixel belongs to a known class.

In the following section, we will demonstrate that maintaining the independence of class scores and constraining them to a known range (*i.e.* $[-1, 1]$) through prototype matching is useful for accurately assessing the confidence of a model in its predictions, and improves the ability to identify anomalous pixels.

2.3.2 Prototopycal Anomaly Segmentation: PAnS

In the previous section, we discussed how the approach of MSP may fail in identifying anomalies due to the softmax normalization, which discards information about the model's confidence. Therefore, we propose a different approach, arguing that it is critical to consider separately the confidence of each class. Ideally, we would like to obtain confidence values that: i) are independent for each class, ii) do not require additional computation, and iii) are bounded within a certain range, such that a threshold for detecting anomalies can be defined on their scores.

To accomplish this, we propose using a prototype to represent each class. Each class prototype can be considered as a reference feature vector for a particular class. We can then compute confidence scores independently for each class by calculating the similarity between the features of any pixel and the prototype. Among the

different ways to define class prototypes in the literature [58–62], we take inspiration from few-shot classification learning works [60, 59] and use a simple but effective cosine classifier, which encodes the class prototypes implicitly in its classification weights.

Cosine Classifier. To efficiently extract class-prototypes, we employ a cosine classifier, which computes the cosine similarities between the input features and the class weights and uses these values as the class scores. While this classifier has previously been utilized for image classification [59, 63–65] to efficiently learn class-prototypes, we are the first to utilize this classifier specifically for the purpose of identifying anomalies in semantic segmentation. In our framework, therefore, we substitute the standard convolutional classifier with a cosine similarity-based classifier.

In particular, given an image x and a pixel i, the classification score for a class c is computed as follows:

$$s_i^c = \rho_c^i(\omega(x)) = \langle \omega_i(x), w_c \rangle = \frac{\omega_i(x)^\intercal w_c}{||\omega_i(x)||\,||w_c||}, \tag{2.2}$$

where $\omega_i(x)$ is the output of the feature extractor ω at pixel i of the image x, and $w_c \in \mathbb{R}^d$ is the prototype of class c. We note that the resulting scores s are in the range $[-1,1]$, due to the normalization term in the denominator.

To learn the prototypes, we use the standard cross-entropy loss on probabilities computed from the scores s^c by means of the softmax function:

$$\ell_{CE}(x,y) = -\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \log \frac{e^{\tau s_i^{y_i}}}{\sum_{c\in\mathcal{C}} e^{\tau s_i^c}}, \tag{2.3}$$

where τ is a scalar value that scales the classification scores to the range $[-\tau,\tau]$ and y_i is the label of pixel i. Intuitively, minimizing the ℓ_{CE} loss helps to ensure that the prototype weights for each class have a low cosine distance with the features of their respective class, effectively representing them on average. In fact, the loss is minimised only when the prototypes are similar to the features set of their corresponding class, and dissimilar from the features set of the other classes. Furthermore, since the feature vectors of a class are pushed closer to the prototype of that class, the features extractor ω is forced to generate features vectors with very low intra-class variance.

This further improves the networks in being more confident when it encounters pixels of known classes and much less confident with anomalous pixels.

Computing the Anomaly Scores. After defining the prototypes, we now introduce how to effectively use them to segment anomalies in the input test images.

To overcome softmax function limitations, we argue that it is important to use the classification scores directly, rather than normalized probabilities. The cosine classifier makes it possible to use class scores s as a measure of the network's confidence in the presence (or absence) of a certain class since they represent the similarity between each class weights and the visual features extracted from the network itself. Additionally, since the scores of the classifier are bounded in the range $[-1,1]$, we can define the binary probability $\bar{\sigma}$ of a class c appearing at pixel i of an image x as:

$$\bar{s}_i^c = \frac{s_i^c + 1}{2} \tag{2.4}$$

and the anomaly score σ using the maximal binary probability, which we define as:

$$\sigma_i(s_i) = 1 - \max_{c \in \mathcal{C}} \bar{s}_i^c. \tag{2.5}$$

Intuitively, σ_i will produce scores close to 1 when the visual features are far from all the class-prototypes, and scores close to 0 if at least one prototype is close to them.

With this approach, we expect our method to effectively represent the known classes, resulting in high confidence for pixels belonging to them. On the other hand, we expect that no class prototype will be close to the features extracted from pixels of anomalous objects. Additionally, by avoiding the aforementioned issues of the softmax function, we can largely boost the results, as we will demonstrate in the following experimental section.

2.3.3 Experiments

In this section, we present experimental settings, baselines, metrics and protocols used to assess the performance of our method, PAnS, in comparison to state-of-the-art approaches for anomaly segmentation. We also conduct an ablation study on the anomaly scores and the classifiers, and provide some qualitative results.

Method	AUPR ↑	AUROC ↑	FPR95 ↓
AE [55]	2.2	66.1	91.7
Dropout [46]	7.5	69.9	79.4
MSP [23]	6.6	87.7	33.7
MSP + CRF [2]	6.5	88.1	29.9
SynthCP [22]	**9.3**	88.5	28.4
PAnS	8.8	**91.1**	**23.2**

Table 2.1 Results under AUPR, AUROC and FPR95 metrics on StreetHazards dataset [2].

Dataset and baselines. Our experiments are conducted on the StreetHazards dataset [2], which is a synthetic dataset for anomaly segmentation proposed within the CAOS benchmark [2] (see Chapter A). We compare our method to several state-of-the-art approaches for anomaly segmentation, including MSP [23], MSP + CRF [2], an auto-encoder (AE) based approach [55], Dropout [46], and the generative approach SynthCP [22].

Metrics. Following previous work [2, 22, 24], we use three anomaly segmentation metrics: AUPR, AUROC, and FPR95 which are commonly used in out-of-distribution detection scenarios [66, 49] as well. AUPR measures the area under the Precision-Recall curve, AUROC measures the area under the True Positive Rate (TPR) and False Positive Rate (FPR) curve, and FPR95 measures the FPR at 95% of recall. For all of these metrics, pixels corresponding to anomalies are considered positive, and all other pixels are considered negative.

Implementation details. Following [22], we use a ResNet-50 architecture [67] as the backbone and PSPNet [26] as the head module of our model. We train the segmentation module for 40 epochs with a batch size of 2 and a learning rate of 0.007. The used learning rate decay policy is a polynomial schedule with a power of 0.9 and a weight decay of 0.0001. We also use InPlace-ABN [68] which allows to save up to 50% of GPUs memory. Similar to [22], we use multi-scale evaluations at test time and perform random scale, random crop, and random horizontal flip augmentations during training.

Comparison with the state of the art. The results of our comparison with state-of-the-art approaches are shown in Table 2.1. As the table reports, our method consistently achieves the best performance by a margin under the AUROC and FPR95 out-of-distribution metrics, while being comparable under AUPR values. Among

all, noteworthy is the result of 23.2% under the FPR95 metric, which indicates that our method is less likely to confuse pixels of known classes as anomalies. This can be attributed to the fact that our prototype-based classifier better preserves the original scores for known classes, which might be either smoothed by the softmax normalization (as in MSP) or overwritten by inaccurate generations (as in SynthCP).

Indeed, under the FPR95 metric, our approach surpasses the previous state-of-the-art (SynthCP) by almost +6%. This increased robustness of our prototype-based classifier against the misclassification of known class pixels is also reflected in the other metrics. Our approach achieves an AUROC of 91.1%, improving upon the previous best method (SynthCP) by 2.6%. These results confirm that our approach achieves the best trade-off between accurately identifying anomalous pixels while preserving at the same time high confidence predictions for pixels of known classes. SynthCP obtains, on the other hand, a slightly better AUPR (+0.5 compared to PAnS). However, it is worth noting that our method only requires a single forward pass on the network, without the need of any generative step and without increasing the computation required by the model.

It is also worth noting that generative models may be affected by the quality of the generated images. In fact, exploiting synthetic images often introduces artifacts [22] that may hinder the performance of generative AS models, by causing them to wrongly segment the artifacts as anomalies (see Fig. 2.2).

Qualitative results. To analyze the impact of our cosine classifier and scores, we provide qualitative examples in Fig. 2.3 comparing the anomaly scores produced by the softmax-based approach MSP [23] and ours for randomly chosen samples from StreetHazards dataset. Higher anomaly scores are represented by white regions, while lower ones are represented by blue regions. As shown in the figure, PAnS is capable of accurately assigning low scores to regions with anomalies (*i.e.*, the *helicopter* in the top image and the *carriage* in the bottom image), while MSP identifies only small portions of the anomalies. However, both approaches have a tendency to assign high anomaly scores to regions where boundaries between known classes can be found, such as the *street lines* and *road* in the top or the *building* and *sidewalk* in the bottom. We believe that modeling these highly uncertain regions between known classes is an open challenge for anomaly segmentation algorithms, which would be important to address in future research.

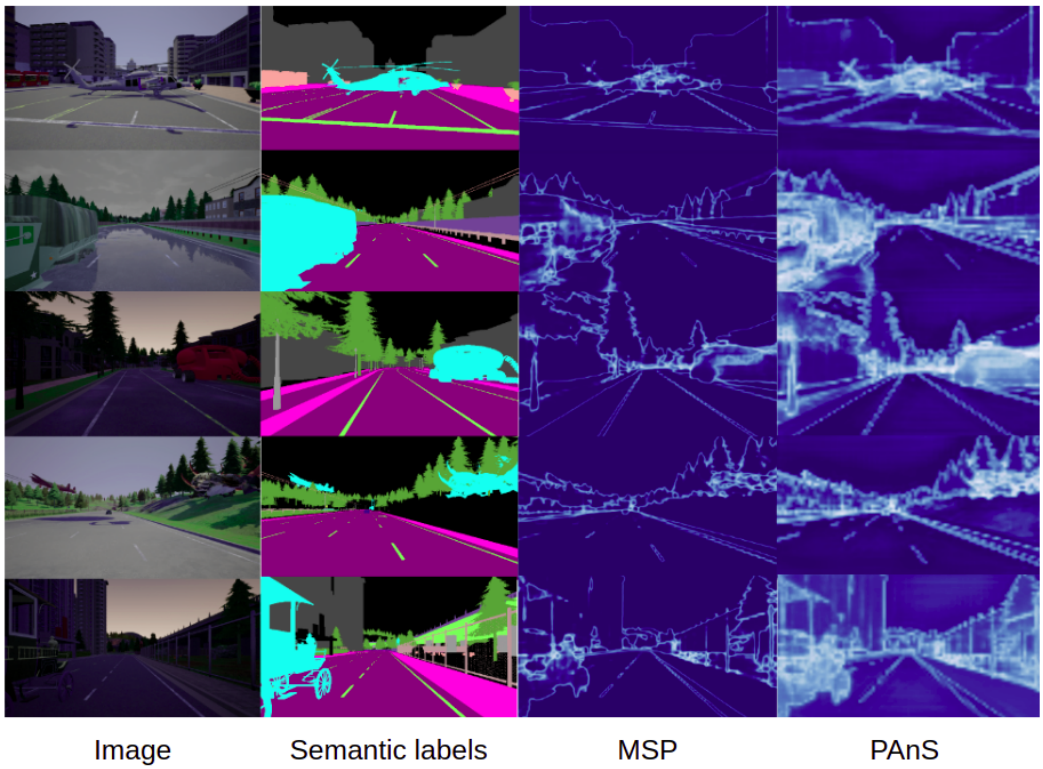


Fig. 2.3 **Qualitative evaluation** of the propability-based approach of MSP and our direct scores while segmenting anomalies on StreetHazards dataset [2]. White pixels indicate a high anomaly score, while the blue ones indicate a low score. Anomalies are represented in cyan in the semantic labels.

Ablation study of anomaly scores. Fig. 2.4 presents an ablation study about the impact of different anomaly score functions σ on StreetHazards [2]. To provide a comprehensive comparison, we considered four variants: i) softmaxed predictions produced by a standard *linear* classifier (MSP [23]); ii) softmaxed predictions produced by a *cosine* classifier (*Cosine cls + softmax*); ii) unnormalized class scores computed by a *linear* classifier (*Class scores*); iv) unnormalized *cosine* scores of PAnS. As shown in the figure, using a cosine classifier to compute the softmax probabilities improves performance compared to a standard classifier, increasing the FPR95 value by 5.8%.

However, directly using the class scores produced by the network instead of the softmax-normalized probabilities is highly beneficial, increasing the standard softmaxed version and the cosine one by 9.5% and 4.7%, respectively. Finally, we note that using the unnormalized cosine scores of our approach outperforms the

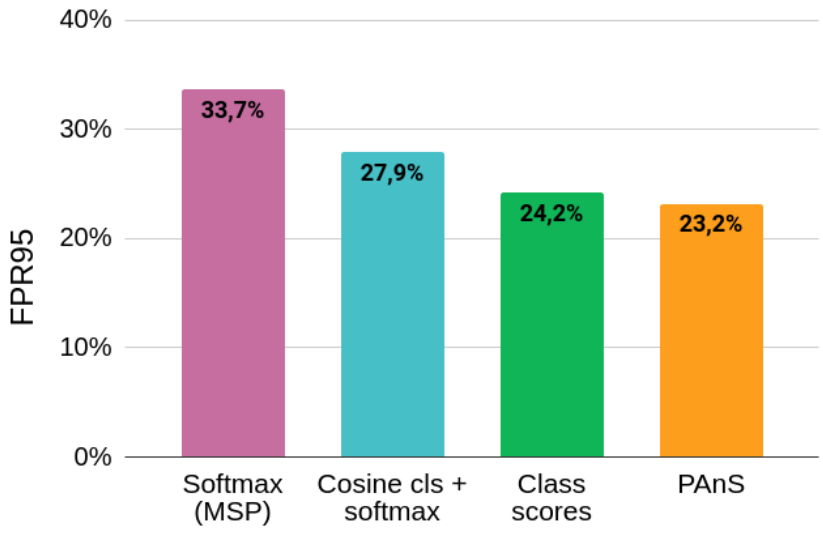


Fig. 2.4 Ablation study on the direct usage of scores produced by both a standard and a cosine-based classifier. Results are computed on StreetHazard dataset [2].

Classifier	bkg	building	fence	pole	street-line	road	sidewalk	veget.	car	wall	t.sign	mIoU
Standard	84.5	70.9	30.1	**23.6**	**26.7**	92.1	57.4	75.1	53.3	42.9	28.9	53.2
Cosine-based	**84.8**	**72.1**	**30.9**	22.3	**26.7**	**92.5**	**60.0**	**75.3**	**55.2**	**45.7**	**30.3**	**54.2**

Table 2.2 Comparison on IoU using a standard and the cosine classifier.

usage of standard class scores, achieving the highest FPR95 value of 23.2%. This improvement can be attributed to the unbounded nature of scores of a standard classification layer, which makes it difficult to define threshold values for detecting anomalies.

Ablation study of classifiers. While our model achieves promising results on AS, an open question is whether it maintains the strong discrimination capabilities of a standard classifier. In Table 2.2, we compare the IoU achieved by both a standard and a cosine-based classifier on the classes of StreetHazards. Overall, the cosine-based classifier performs better than the standard one, achieving a 54.2% mIoU compared to a 53.2% mIoU for the standard classifier. The results show that the cosine similarity allows reaching higher mIoU values on almost every class, especially on those that are typically considered difficult, such as fence, sidewalk, and traffic sign. The only exception is the *pole* category, where the performance of the model slightly decreases to 22.3% compared to 23.6% for the standard classifier. The cause of this behaviour

is the small size and infrequent representation of the *pole* class in the StreetHazards dataset, which makes it challenging for the model to estimate a good prototype for it.

Limitations. As depicted in Figure 2.3, an unresolved issue remains the uncertainty that the model has on the pixels located at the boundaries between different semantic classes. Despite achieving better performance than traditional classifiers, out model still struggles to provide highly confident predictions for these particular regions. The reason for this limitation lies in the fact that pixels situated on boundaries often differ from those that represent the semantics of a specific class. Consequently, they deviate significantly from the expected values encoded in the prototypes. As a result, the prototypes end up encoding features that are dissimilar from those found at the boundaries and, due to this discrepancy, the model sometime fails to recognize these pixels and categorizes them as anomalies. It would be interesting to investigate if transformer-based architectures [38, 40] would be able to mitigate this issue given their ability to incorporate semantic context.

2.3.4 Conclusions

In this section, we addressed the issue of anomaly detection in semantic segmentation, namely anomaly segmentation, which is an important yet scarcely studied topic. Previous approaches have either concentrated on modeling the probability of a pixel belonging to an unknown category or on generative methods for detecting anomalies via inconsistencies in their reconstructions. In contrast, we argue that measuring the distance between the visual features extracted from a pixel and a general representation of each class is a more effective approach than relying on maximum softmax probabilities for identifying that pixel as anomalous. To this end, we use class-specific prototypes extracted from the weights of a cosine similarity-based classifier to learn such general representations. Experimental results on the widely used StreetHazards benchmark show that our approach outperforms previous state-of-the-art methods in two of the three metrics commonly used in anomaly segmentation literature by a significant margin.

Chapter 3

Learning new semantic concepts

Existing semantic segmentation models achieved over the years impressive results in a variety of applications, but they still struggle to update their knowledge over time, which limits their adoption in real world environments. Moreover, pixel-wise annotations are expensive and time-consuming to obtain. In this chapter we propose a framework for Weakly Incremental Learning for Semantic Segmentation, which aims to segment new classes from cheap image-level labels. We start by introducing the problem formulation in Section 3.3.1 and a related work review in Section 3.2. In Section 3.3 we present our method, that differently from existing approaches that generate pseudo-labels offline, uses a localizer module trained with only image-level labels to obtain pseudo-supervision online used to update the segmentation model over time. Furthermore, we propose a way to exploit the localizer-generated soft-labels to reduce the noise generated in the process. Experimental results on the Pascal VOC and COCO datasets (Section 3.3.4) show that our approach outperforms offline weakly-supervised methods and achieves results comparable to fully supervised incremental learning methods.

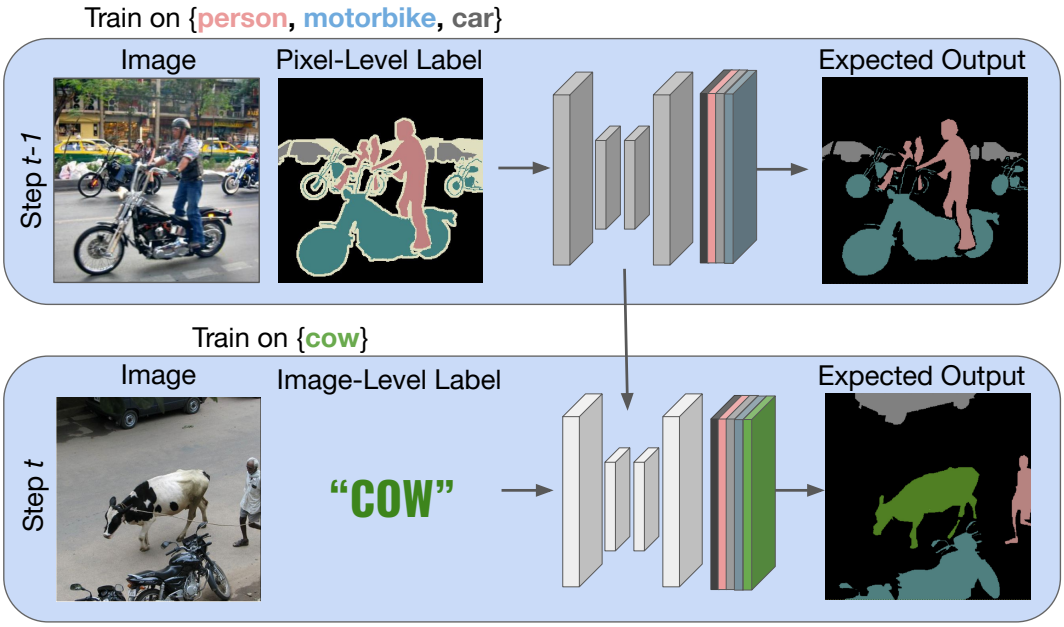


Fig. 3.1 Overview of Weakly-Supervised Incremental Learning for Semantic Segmentation (WILSS). We start by considering a model pre-trained on a set of categories (*e.g.*, *person, motorbike, car*), using expensive pixel-wise annotations. Then, the model is incrementally updated to segment new categories (*e.g.*, *cow*) using only image-level annotations and without having access to old data.

3.1 Problem statement

Despite the significant progress achieved in recent years in semantic segmentation, especially thanks to the rapid growth of deep learning approaches [20, 30, 25] and the availability of large-scale annotated datasets [69–73], we highlighted in Chapter 1 how crucial is to make models able to update over time their internal knowledge to successfully operate in the real world. A naïve approach would be annotating new samples and adding them to existing datasets, in order to train new models from scratch. However, this approach is not feasible when updates are frequent, as training on the entire augmented dataset would require too much time, increasing carbon footprint and energy consumption of machine learning models [74–76]. In addition, retraining or fine-tuning the models becomes infeasible when the original data is no longer accessible, for instance due to privacy or intellectual property concerns.

In this chapter we focus on the challenge known as class-*incremental learning (IL)* [77–81], which involves incrementally updating pre-existing models when new classes are available, as proposed in several recent works [41, 44, 15, 42, 14].

These IL approaches update models through multiple learning steps, where each step contains only the newly available data, and they employ ad-hoc techniques to mitigate the catastrophic forgetting issue [82]. While these approaches reduce the cost of training, they still require pixel-wise supervision on new classes, which is often expensive and time-consuming to collect, and typically requires expert human annotators [70, 83].

To reduce the cost of annotations, various forms of weak supervision have been proposed, such as points [84], scribbles [85, 86], bounding boxes [87, 88] and image-level labels [89–91]. Image labels, in particular, can be easily obtained from image classification benchmarks [92] or from the web, dramatically reducing the annotation cost. Nevertheless, their use in an incremental learning setting has not been previously explored.

To this end, in this chapter we propose to incrementally train semantic segmentation models using only image-level labels for the new classes and we name this task *Weakly-Supervised Incremental Learning for Semantic Segmentation* (WILSS). This new setting combines the advantages of incremental learning (avoiding an entire re-training phase) with those of weak supervision (cheap and widely available annotations). An illustration of WILSS is shown in Fig. 3.1.

Directly applying existing weakly-supervised segmentation strategies to incremental segmentation ones would involve (i) extracting pixel-level pseudo-supervision offline using a weakly-supervised approach [4, 3, 93, 94, 5] and (ii) updating the segmentation network using an incremental learning technique [44, 15, 14]. However, we argue that creating offline pseudo-labels in incremental settings is sub-optimal, as it requires two separate training stages, and it prevents exploiting the model's knowledge on previous classes to learn more efficiently the new ones.

Therefore, we propose a **W**eakly **I**ncremental **L**earning framework for semantic **S**egmentation, which incrementally trains a segmentation model by means of **ON**line pseudo-supervision from image-level annotations **(WILSON)** and leverages previous knowledge to learn novel classes.

We propose to extend the standard encoder-decoder semantic segmentation architecture [20, 30, 25] by introducing a new module on the encoder, the *localizer*. Its role is to generate pixel-level pseudo-supervision for the segmentation backbone starting from image-level labels. To improve the pseudo-supervision, we guide the training of the localizer with a pixel-wise loss originated from the predictions of

the segmentation model. This regularization serves two main purposes: (i) it serves as a strong prior for the previous class distribution, helping the model to locate old classes within the image, and (ii) it provides a saliency prior for the extraction of more accurate object boundaries. To mitigate the issue of noise in the pseudo-supervision, we do not use hard pseudo-labels as in previous approaches ([3–5]), but rather obtain soft-labels from the localizer, which provides information about the probability of a pixel belonging to a specific class.

Contributions. To summarize, the contributions presented in this chapter are the following: i) we present WILSS, the Weakly supervised Incremental Learning for Semantic Segmentation task in which we extend pre-trained segmentation models by adding new classes using only image-level supervision; ii) we present WILSON, a novel framework which generates pseudo-supervision online by means of a localizer trained with two loss functions: an image-level classification loss function and a pixel-wise localization loss function. To account for noise in the pseudo-supervision, we use a convex combination of soft and hard labels, which leads to improved segmentation performance compared to using hard labels only; iii) we demonstrate the effectiveness of the proposed method through evaluations on the Pascal VOC [69] and COCO [70] datasets, showing outstanding performance compared to offline weakly-supervised methods and comparable or slightly inferior performance compared to fully supervised incremental learning methods.

3.2 Literature review

Incremental learning. While several attempts to address this task have been made over the years for shallow models, e.g. [95, 96], the field has lately experienced a rise in interest for deep learning models due to the additional challenges they provide. Existing approaches for deep architectures fall mostly into three categories: parameter isolation-based [97–99], regularization-based (prior- and data-focused) [100, 101, 79, 77, 102] and rehearsal-based [78, 80, 103, 81, 104, 105]. Parameter isolation-based approaches allocate a parameters subset to each task, and inhibit their modification through the subsequent learning stages to prevent forgetting. Similarly, prior-focused regularization-based strategies [79, 101, 100, 106] constraining the learning of new tasks by penalizing changes of important old parameters, rely on knowledge stored in parameters value. In contrast, data-focused ones [77, 102, 107–110, 81] make use of the distillation paradigm [111] by applying a regularisation term based on the distance between the activation computed by the network at the current step and at the previous one. Lastly, rehearsal-based strategies adopt examples belonging to prior tasks, which are either generated [103–105] or stored [78, 80, 81, 112], and used to prevent the model from forgetting them during the training phase of the following task.

Even though it has been extensively investigated in image classification [78, 98, 77, 81, 80, 104, 97, 99] and object detection [113–121], IL is still in the early stages in semantic segmentation [41, 44, 15, 42, 14]. [44] has been the first to identify the background shift problem as the cause of catastrophic forgetting in segmentation. The authors modified the cross-entropy loss such that only the probability of old classes was propagated across the incremental steps, and they implemented a novel distillation term to maintain previous knowledge. Later on, [15] proposed to maintain long and short range relationships in the feature space; while [42] improved class-conditional latent separations by regularizing the features space. Slightly closer to our task is the work of [14], which pseudo-labels downloaded images from the web to integrate samples of old classes during the learning stages. However, while they use class labels to identify images of old classes, in section Section 3.3 we focus on learning novel categories that have never been encountered before, relying solely on cheap image-level labels.

Weakly supervised semantic segmentation. Collecting pixel-wise annotation to supervise semantic segmentation models is typically a costly and time-consuming process. The goal of the weakly supervised semantic segmentation task is to build effective semantic segmentation models utilising only cheap annotations, that generally come in the form of bounding boxes [87, 122, 88], scribbles [123, 86], points [83, 124], and image-level labels [125, 126, 91, 94].

This thesis focuses on image-level supervision, which has garnered more interest than other forms of weak supervision due to its low cost and wide adoption. The majority of image-based weakly supervised methods [91, 127, 126, 93, 128, 125, 94, 129] adopt a two-step procedure: (i) starting from the image-level annotations, they generate pixel-level pseudo-labels, which are then (ii) used to train a segmentation backbone. To extract the pseudo-labels, [130] proposed to use the Class Activation Maps (CAMs) computed from an image-level classifier. Subsequent works mainly focused on improving the pseudo-annotations through refinements steps [93, 128], additional losses [91, 126, 129, 3, 94, 4], or by erasing portions of the images [131–133] which makes the CAM expanding also to non-non-discriminatory image areas. Moreover, [5, 134] proposed to use additional external supervision, e.g. saliency, to improve the pseudo-labels object boundaries. The only exception is [4], which trains in one single stage a segmentation model directly from the image-level labels.

Despite the fast development of strategies for generating pseudo-labels from image-level supervision, these works only operate in a static environment in which the model is limited to learn from a predetermined set of classes. Instead, in section Section 3.3 we concentrate on the more difficult incremental learning scenario, in which additional classes are learned over time using only cheap image-level labels.

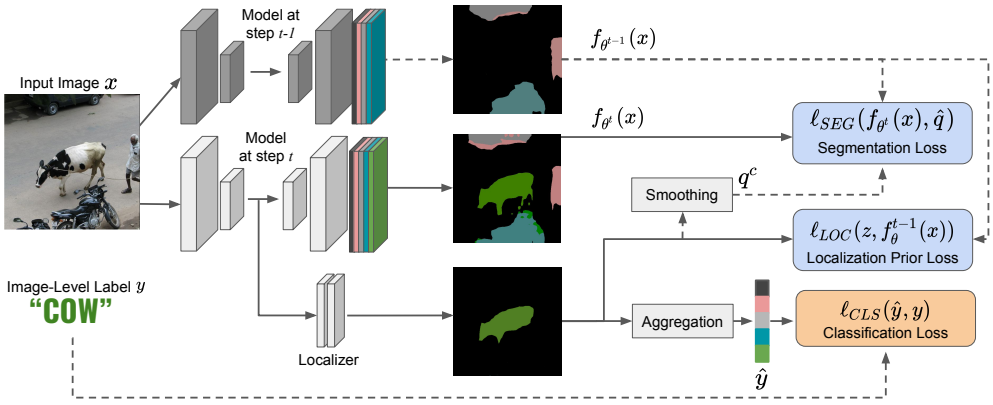


Fig. 3.2 Illustration of the end-to-end training of WILSON. The localizer module is directly trained using ℓ_{CLS} (the classification loss) and ℓ_{LOC} (the Localization Prior loss) which exploits prior knowledge derived from the old model at step $t-1$. The supervision of the segmentation model, on the other hand, comes from both CAM and old model output. The dotted lines indicate no backpropagation of the gradient.

3.3 Incremental learning in semantic segmentation from image labels

Adapting current WSSS methods [3, 5, 135, 93, 91] for incremental segmentation requires generating pseudo-labels offline for the novel classes, and training a separate segmentation model after. Instead, we present an end-to-end framework for WILSS that allows incremental learning from pseudo-labels by means of a localizer attached to the model. The details of the framework are outlined as follows: in Section 3.3.1 we define the problem and notation; in Section 3.3.2 we illustrate how to train the classification to obtain pseudo-supervision, and in Section 3.3.3 we describe how the segmentation model is trained to learn new classes without forgetting the old ones. The overall structure of the framework is illustrated in Fig. 3.2.

3.3.1 Problem formulation

We define our input space $\mathcal{X}$ as the image space, and we assume that each image is composed of a set of pixels $\mathcal{I}$ with a fixed cardinality $|\mathcal{I}| = H \times W = N$. We then define the output space $\mathcal{Y}^N$ as the product set of N-tuples with elements in a label space $\mathcal{Y}$. In standard semantic segmentation settings, given an image $x \in \mathcal{X}$, the goal is to learn a mapping function that assigns each pixel x_i to a label $y_i \in \mathcal{Y}$, representing

its semantic class. This function is realized by a model $f_\theta = d_{\theta^d} \circ e_{\theta^e} : \mathcal{X} \mapsto \mathbb{R}^{N \times |\mathcal{Y}|}$ mapping images in the image space $\mathcal{X}$ to a pixel-wise class probability vector. We denote the encoder and the decoder of the segmentation networks with e and d, respectively.

The output segmentation mask, denoted as $y^* = \{\arg\max_{c \in \mathcal{Y}} p_i^c\}_{i=1}^N$, is obtained by taking for each pixel the class with the highest probability, where p_i^c represents the model's prediction for pixel i belonging to class c. In the incremental semantic segmentation scenario [44], training occurs over multiple *learning steps*. At each step t, the previous set of labels $\mathcal{Y}^{t-1}$ is augmented with new classes $\mathcal{C}^t$, resulting in a new set of labels $\mathcal{Y}^t = \mathcal{Y}^{t-1} \cup \mathcal{C}^t$.

Differently from the original incremental segmentation setting, in WILSS, we are only provided with dense annotations for the initial step ($t = 0$). This means that the model is initially pre-trained on a densely-annotated dataset $\mathcal{T}^0 \subset \mathcal{X} \times (\mathcal{C}^0)^N$ containing only the initial classes. For all subsequent steps ($t > 0$), we only have access to training sets with cheap image-level annotations for the novel classes $\mathcal{T}^t \subset \mathcal{X} \times (\mathcal{C}^t)$. Following [44], we assume that data belonging to previous training steps is no longer available, and we want to extend our model to perform segmentation on new classes while preserving its performance on old classes *i.e.* $f_{\theta^t} : \mathcal{X} \mapsto \mathbb{R}^{N \times |\mathcal{Y}^t|}$.

3.3.2 Training the Localizer

Inspired by previous work on WSSS [4, 3, 5, 135, 93, 91], we introduce a *localizer* g which is trained with image-level labels to generate pseudo-supervision for the segmentation model. The localizer takes in the features from the segmentation encoder e and produces a score for each class (including background, old, and new classes) *i.e.* $z = g(e(x)) \in \mathbb{R}^{|\mathcal{Y}^t| \times H \times W}$.

Learning from image-level labels. To learn from image-level labels, we need to firstly aggregate the pixel-level classification scores z. A common approach is to simply aggregate the features $e(x)$ via Global Average Pooling (GAP) [93, 3] and classify them afterwards, but this leads to coarse pseudo-labels [4] as all the pixels in the feature map are encouraged to identify with the target class, resulting in less discriminative learned features.

Instead, following [4], we use the *normalized Global Weighted Pooling* (nGWP), which avoids a direct aggregation on the features, but rather weights each pixel based on its relevance for the target class. Specifically, the weight of each pixel is computed with the `softmax` operation ψ, *i.e.* $m = \psi(z)$, and the final aggregated scores are computed as:

$$\hat{y}^{nGWP} = \frac{\sum_{i \in \mathcal{I}} m_i z_i}{\varepsilon + \sum_{i \in \mathcal{I}} m_i}, \tag{3.1}$$

where ε is a small constant. Furthermore, to encourage the scores to identify all the visible pieces of the object, we also employ the *focal penalty* term introduced by [4], which is calculated as:

$$\hat{y}^{FOC} = (1 - \frac{\sum_{i \in \mathcal{I}} m_i}{|\mathcal{I}|})^{\gamma} log(\lambda + \frac{\sum_{i \in \mathcal{I}} m_i}{|\mathcal{I}|}), \tag{3.2}$$

in which λ and γ are hyper-parameters. When the weights $\frac{\sum_{i \in \mathcal{I}} m_i}{|\mathcal{I}|}$ are non-zero, $p > 0$ discounts further increases, to focus on the failure cases of weights that are, instead, near-zero. We invite the readers to consult [4] for further details on nGWP and the focus penalty.

Being WILSS an incremental learning scenario, we assume to only have access to image-level annotations y for the *new classes* $\mathcal{C}^t$. We then train the localizer minimizing the widly adopted *multi-label soft-margin loss* [93, 3, 4]:

$$\ell_{CLS}(\hat{y}, y) = -\frac{1}{|\mathcal{K}|} \sum_{c \in \mathcal{K}} y^c log(\hat{y}^c) + (1 - y^c) log(1 - \hat{y}^c), \tag{3.3}$$

where $\mathcal{K} = \mathcal{C}^t$, $\hat{y} = \sigma(\hat{y}^{nGWP} + \hat{y}^{FOC})$, and σ is the logistic function.

It is important to note that, although the loss is computed only on the new classes, it implicitly depends on the scores of the old classes as well, due to the softmax-based aggregation in Eq. (3.1). However, given the low cost of image-level annotations and the ease with which new images can be weakly annotated, we may also take into consideration a more relaxed setting, with weak annotations provided for both old and new classes. In this scenario the classification loss in Eq. (3.3) is calculated on all classes and $\mathcal{K} = \mathcal{Y}^t$.

Localization Prior. The image-level labels only indicate the presence of new classes in the image, but do not provide any information about their boundaries or any insights into the location of old classes. However, we argue that these cues can

be freely obtained from the segmentation model learned in previous training steps. In particular, we can use the background score as a saliency prior to extract more accurate object boundaries. Additionally, the scores of the old classes can be used to guide the localizer in detecting where and whether old categories are present in the image, directing its attention to alternative regions.

Therefore, we introduce on the localizer a direct signal of supervision coming from the segmentation model learned in step $t-1$, denoted as f_θ^{t-1}. This supervision acts as a *Localization Prior* (LOC) and is provided in the form of a pixel-wise loss between the segmentation model outputs $\xi = \sigma(f_\theta^{t-1}(x))$ and the classification scores z. The objective function to be minimized is as follows:

$$\ell_{LOC}(z,\xi) = -\frac{1}{|\mathcal{Y}^{t-1}||\mathcal{I}|}\sum_{i\in\mathcal{I}}\sum_{c\in\mathcal{Y}^{t-1}} \xi_i^c log(\sigma(z_i^c)) + (1-\xi_i^c) log(1-\sigma(z_i^c))), \quad (3.4)$$

where $\sigma(\cdot)$ is the logistic function.

In Eq. (3.4), the segmentation model provides dense target for old classes. Differently from the `softmax` operator, which forces competition among classes, the `logistic` function allows class probabilities to be independent. This is beneficial for an accurate localization prior, as in the case of a new class low scores for both old classes and the background will be produced, implicitly indicating to the localizer that the pixel belongs to a novel category.

3.3.3 Learning to Segment from Pseudo-Supervision

In order to train a semantic segmentation network, WSSS methods often extract hard-pseudo labels from an image-level classifier. These labels are created by generating a one-hot distribution $q^{\text{Hard},c}$ for each pixel, where the class with the highest score is given a value of one and all other classes are given a value of zero, *i.e.*

$$q_i^{\text{Hard},c} = \begin{cases} 1 & \text{if } c = \arg\max_{k\in\mathcal{Y}^t} m_i^c \\ 0 & \text{otherwise} \end{cases} \quad (3.5)$$

where m is the softmax normalized score obtained from the localizer.

However, pseudo-supervision generated from image-level classifiers is well-known for being noisy [135, 5, 4, 3], and using $q^{H,c}$ to supervise the segmentation network might be deleterious for learning, resulting in the model fitting the wrong targets. To address this issue, we propose smoothing the pseudo-labels to reduce the noise [136]. Formally, the pseudo-supervision q^c for a given class c is computed as:

$$q^c = \alpha q^{\text{Hard},c} + (1-\alpha)m^c, \tag{3.6}$$

where α is a hyper-parameter that controls the smoothness of the pseudo-labels.

Although the localizer generates scores for both new and old categories, the output distribution might be biased in favor of new categories due to the incremental training step. This can lead to catastrophic forgetting [82] if we use q as the target for the segmentation model. Inspired by the knowledge distillation framework [111], we replace the pseudo-supervision extracted from the localizer on old classes with the output of the segmentation model trained in the preceding learning step. The final pixel-level pseudo-supervision $\hat{q}$ is defined as follows:

$$\hat{q}^c = \begin{cases} \min(\sigma(f_{\theta^{t-1}}(x))^c, q^c) & \text{if } c = \texttt{b}, \\ q^c & \text{if } c \in \mathcal{C}^t, \\ \sigma(f_{\theta^{t-1}}(x))^c & \text{otherwise}, \end{cases} \tag{3.7}$$

where $\texttt{b}$ represents the background class and $\sigma(\cdot)$ is the logistic function. We note that we use the minimum value of the two distributions for the background class, which helps to model the background shift [44].

Nevertheless, the pseudo-supervision $\hat{q}^c$ is not a probability distribution that sums to one, which is required, instead, by the standard softmax-based cross-entropy loss. Therefore, we propose to use the multi-label soft-margin loss as the training loss:

$$\ell_{SEG}(p,\hat{q}) = -\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\sum_{c\in\mathcal{Y}^t} \hat{q}_i^c log(p_i^c) + (1-\hat{q}_i^c) log(1-\sigma(p_i^c)), \tag{3.8}$$

where $\mathcal{Y}^t$ is the set of all seen categories and $p = f_{\theta^t}(x)$ is the output of the segmentation model.

In conclusion, it is worth noting that the localizer is not used during the testing phase, thus our method does not increase the time required for inference.

3.3.4 Experiments

In this section, we present experimental settings, baselines, and protocols used to assess the performance of our method, WILSON, in the novel WILSS task against weakly supervised semantic segmentation approaches and fully-supervised incremental segmentation methods. Moreover, we present three ablation studies evaluating the effectiveness of the localization prior, the smoothing effects on pseudo-labels and the effect of having available supervision for both old and new classes in the incremental steps.

Datasets and Settings. We extensively evaluate WILSON on the Pascal VOC 2012 [69] and COCO [70] standard segmentation benchmarks (see Chapter A). We adopt the training split and annotation of [137] which resolves the problem of overlapping annotations in [70].

Following prior works [44, 14], we conducted experiments on the Pascal VOC dataset in two incremental learning settings: the **15-5 VOC**, in which 15 categories are learned in the first learning stage, and 5 new categories are added in the second step; and the **10-10 VOC**,in which two phases of 10 categories each are performed. As in [44, 14], we report results in two experimental protocols: (i) the *disjoint* scenario, where only images containing new or previously seen categories are included in each training step; and (ii) the *overlap* scenario, where each training stage contains all the images having at least one pixel from a new class. In addition, we propose a new incremental learning scenario called **COCO-to-VOC**, which consists of two training steps. In the first step, we learn the 60 COCO categories that are not present in the Pascal VOC dataset, excluding all the images containing at least one pixel from Pascal VOC categories. In the second step, we then learn 20 Pascal VOC classes. As in previous protocols [44, 14], we report results on the dataset validation sets, as the test set labels are not publicly available. To evaluate the performance of the segmentation model, we used the standard mean Intersection over Union (mIoU) metric [69].

We recall that, unlike [44, 14], in the proposed WILSS setting, each incremental step provides only image-level labels for the novel cateogires.

Baselines. Since WILSS is a new setting, we compare WILSON with both incremental learning and weakly supervised semantic segmentation strategies. We report the results of eight methods that are currently the state-of-the-art for incremental learn-

ing using pixel-wise supervision: LWF [77], LWF-MC [78], ILT [41], CIL [138], MiB [44], PLOP [15], SDR [42], and RECALL [14]. Note that RECALL [14] uses additional images from the Web, which is not the case for the other methods. For the Pascal VOC dataset, we use the results published in [14, 15], and for the COCO-to-VOC setting, we run the experiments using the code provided by [44].

Moreover, we report the performance of WSSS state-of-the-art methods adapted to act in an incremental learning scenario. Specifically, we first train a classification model on the available images in the incremental learning steps, then we generate hard pseudo-labels offline and we train the segmentation model minimizing the loss in Eq. (3.8). We report results using the pseudo-labels generated from the following methods: class activation maps (CAM) obtained from a standard image classifier, SEAM [3], SS [4], and EPS [5]. As with WILSON, we followed the same experimental protocols of [44], training each method using only the images in disjoint and overlap scenarios. To produce the results, we used the implementation released by the authors of each method. For CAM, we used the same implementation EPS used. It is important to note that while CAM, SS, and SEAM only use image-level labels, EPS also relies on an off-the-shelf saliency detector trained on external data.

Implementation Details. In all our experiments, we employed Deeplab V3 architecture [20] with either a ResNet-101 [67] backbone (for Pascal VOC) or a Wide-ResNet-38 [139] backbone (for COCO), both pre-trained on ImageNet. The ResNet-101 had an output stride of 16, while the Wide-ResNet-38 had an output stride of 8. Following [44], to reduce memory requirements we employed in-place activated batch normalization [68]. The localizer module is made of three convolutional layers, followed by a batch normalization layer and Leaky ReLU. The first two layers have a kernel size of 3×3, while the last one has a kernel size of 1×1, with {256, 256, number of classes} channel numbers, and stride equal to 1. The model was trained for a total of 40 epochs using a batch size of 24. We used SGD with 0.001 as the initial learning rate (0.01 was used for the Deeplab head and the localizer), 0.9 as momentum and 10^{-4} as weight decay. We started by training only the localizer for the first 5 epochs; thereafter, we trained the entire network by adding the pseudo-supervision generated by the localizer, and employing a polynomial schedule with 0.9 as power. As in [4], we set $\lambda = 0.01$ and $\gamma = 3$ in Eq. (3.2), and

		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 ⋆ [77]	Pixel	39.7	33.3	38.2	67.0	41.8	61.0
LWF-MC ⋆ [78]	Pixel	41.5	25.4	37.6	59.8	22.6	51.0
ILT ⋆ [41]	Pixel	31.5	25.1	30.0	69.0	46.4	63.6
CIL ⋆ [138]	Pixel	42.6	35.0	40.8	14.9	37.3	20.2
MIB ⋆ [44]	Pixel	71.8	43.3	64.7	75.5	49.4	69.0
PLOP ◇ [15]	Pixel	71.0	42.8	64.3	<u>75.7</u>	51.7	<u>70.1</u>
SDR ⋆ [42]	Pixel	<u>73.5</u>	47.3	<u>67.2</u>	75.4	52.6	69.9
RECALL ⋆ [14]	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 [3]	Image	71.0	33.1	62.7	68.3	31.8	60.4
SS [4]	Image	71.6	26.0	61.5	72.2	27.5	62.1
EPS [5]	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**

Table 3.1 Results on Pascal VOC 15-5 setting expressed in mIoU%. Best Image-level supervision method is bold. Best Pixel-level supervision method is underlined. ⋆: results from [14]. ◇: results from [15].

after the fifth epoch, we used the self-supervised segmentation loss on the localizer. Lastly, we set in all our experiments $\alpha = 0.5$ in Eq. (3.6).

Single step addition of five classes (15-5). In this setting, the 5 classes of Pascal VOC dataset added after the initial learning stage are: *plant, sheep, sofa, train, tv-monitor*. The results are reported in Table 3.1.

Despite being trained only using image-level annotations, WILSON achieves competitive results against approaches trained with full pixel-wise supervision in all the settings (both disjoint and overlap). Considering all the classes (disjoint scenario), WILSON outperforms SDR [42] by 0.1% and RECALL [14] by 1.0%, demonstrating WILSON's resilience to forgetting without the use of a replay buffer while maintaining sufficient plasticity for learning new classes. Additionally, in the disjoint scenario, we are able to surpass PLOP [15] by 1.0% and MIB [44] by 0.5% on the new classes.

When comparing WILSON to other WSSS methods adapted to the WILSS setting, the results demonstrate the strengths of WILSON: the ability to retain knowledge of past categories and, most importantly, its ability to learn new semantic

		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 ⋆ [77]	Pixel	63.1	61.1	62.2	70.7	63.4	67.2
LWF-MC ⋆ [78]	Pixel	52.4	42.5	47.7	53.9	43.0	48.7
ILT ⋆ [41]	Pixel	67.7	61.3	64.7	70.3	61.9	66.3
CIL ⋆ [138]	Pixel	37.4	60.6	48.8	38.4	60.0	48.7
MIB ⋆ [44]	Pixel	66.9	57.5	62.4	70.4	63.7	67.2
PLOP [15]	Pixel	63.7	60.2	63.4	69.6	62.2	67.1
SDR ⋆ [42]	Pixel	67.5	57.9	62.9	70.5	63.9	67.4
RECALL ⋆ [14]	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 [3]	Image	65.1	53.5	60.6	67.5	55.4	62.7
SS [4]	Image	60.7	25.7	45.0	69.6	32.8	52.5
EPS [5]	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**

Table 3.2 Results on Pascal VOC 10-10 setting expressed in mIoU%. Best Image-level supervision method is bold. Best Pixel-level supervision method is underlined. ⋆: results from [14].

categories using only image-level annotations. Indeed when considering new classes, WILSON outperforms EPS [5] by +5.3% mIoU in the disjoint scenario, even if EPS uses off-the-shelf generated saliency maps as additional supervision. Additionally, we outperform SEAM [3] and SS [4] by 11.7% and 17.8%, respectively. These improvements are even more pronounced in the overlap scenario, where WILSON not only retains all its prior knowledge, but also achieves a +7.2% boost when learning new categories compared to EPS. In this situation, the overall improvement is +5.1% when compared to the best methods (SS and EPS).

Single step addition of ten classes (10-10).

In this setting, we add 10 categories in the incremental step: *dining-table, dog, horse, motorbike, person, plant, sheep, sofa, train, tv-monitor*. As shown in Table 3.2, the results are consistent with the 15-5 setting. The differences between WILSON and pixel-wise supervision are minor and the results are nearly comparable. In terms of mIoU, the gap between WILSON and the most accurate incremental learning methods, *i.e*. ILT in the disjoint scenario and SDR in the overlapped one, is 3.9% w.r.t. the former and shrinks to 2.4% w.r.t. the latter. The efficacy of WILSON is

Method	Sup	COCO 1-60	COCO 61-80	COCO All	VOC 61-80
FT	Pixel	1.9	41.7	12.7	<u>75.0</u>
LWF [77]	Pixel	36.7	<u>49.0</u>	<u>40.3</u>	73.6
ILT [41]	Pixel	<u>37.0</u>	43.9	39.3	68.7
MIB [44]	Pixel	34.9	47.8	38.7	73.2
PLOP [15]	Pixel	35.1	39.4	36.8	64.7
CAM	Image	30.7	20.3	28.1	39.1
SEAM [3]	Image	31.2	28.2	30.5	48.0
SS [4]	Image	35.1	36.9	35.5	52.4
EPS [5]	Image	34.9	38.4	35.8	55.3
WILSON (ours)	Image	**39.8**	**41.0**	**40.6**	**55.7**

Table 3.3 Results on COCO-to-VOC setting expressed in mIoU%. The best method using Image-level supervision is bold. Best Image-level supervision method is bold. Best Pixel-level supervision method is underlined.

also confirmed when compared to WSSS (image-level annotations) methods. Indeed, when learning new semantic classes, our online strategy outperforms all offline competitors by more than $+0.7\%$ overall mIoU in the overlap protocols, while achieving comparable results (+0.2%) in the disjoint scenario. Furthermore, we provide qualitative results confirming the superiority of WILSON on both old and new categories in Fig. 3.4.

COCO-to-VOC. We consider this set of experiments the most challenging. The network is first trained on 60 categories from the COCO dataset (unshared with the Pascal VOC dataset) and 20 classes from the the Pascal VOC dataset are then added in the second step. We report in Table 3.3 the evaluations on both COCO and Pascal VOC validation sets. Despite experiencing a 8% drop in performance compared to LwF when learning new classes, WILSON still demonstrates its ability to retain prior knowledge while learning new categories under image-level supervision surpassing ILT on old classes by $+2.8\%$, with ILT being the top competitor trained with pixel-wise annotations. When comparing against WSSS methods, WILSON outperforms all of them, achieving 4.8% improvements in terms of mIoU over EPS (best WSSS method) on the COCO dataset. Similar results also hold true for the Pascal VOC validation set. WILSON outperforms all previous WSSS methods on both old and new categories, on both COCO and Pascal VOC.

		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**

Table 3.4 Ablation study to validate the robustness of pseudo-supervision evaluating different types of localization priors for training the localizer.

	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 [3]	71.9	26.9	61.7	70.8	28.1	61.0
SS [4]	71.8	26.3	61.7	72.1	27.6	62.1
EPS [5]	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 [3]	66.0	50.4	59.7	70.9	54.6	64.0
SS [4]	60.8	26.0	45.2	69.6	33.0	52.6
EPS [5]	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 3.5 Ablation study to evaluate weakly supervised segmentation methods trained using direct supervision on both old and new classes in the incremental step.

Ablation study on the Localization Prior. To validate the robustness of our pseudo-supervision generation, we conducted an ablation study on the VOC 10-10 disjoint and overlap scenarios considering different strategies for training the localizer. The results are shown in Table 3.4. Specifically, we compared the following strategies: (i) using a constant value for the old classes, as in [4], (ii) using a fixed prior, directly concatenating the old model's segmentation output to the class scores when computing m, (iii) providing a localization supervision to the localizer using the softmax cross-entropy loss, and (iv) using the loss in Eq. (3.4). Overall mIoU achieved by using a constant value and disregarding past knowledge from the old segmentation network is lower when compared to using a localization prior, especially for new classes (-5.1% on overlap and -4.4% on disjoint). This demonstrates that teaching the location of previous classes to the localizer might prevent forgetting and improve

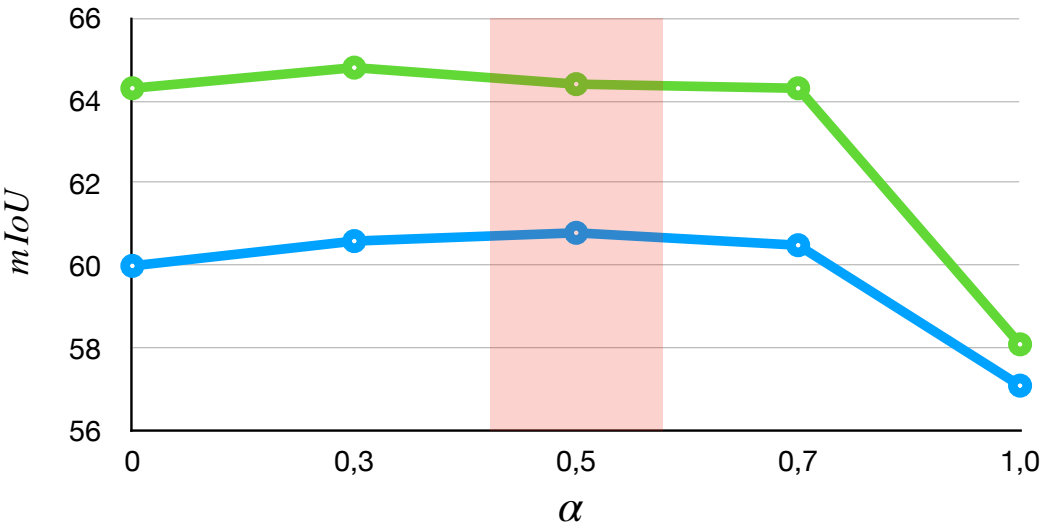


Fig. 3.3 Ablation study on the effect of α on smoothing one-hot pseudo-labels used to supervise the ℓ_{SEG}. Reported mIoU for both the Disjoint and Overlap VOC 10-10 protocols.

performance when learning new categories. Thereby, using aggressive priors, such as the direct segmentation output of the old model, does not allow the network to properly learn the new categories, resulting in a gap of -4.3% on overlapped scenario and -4.0% on disjoint scenario w.r.t. ℓ_{LOC}. Furthermore, using the softmax cross-entropy loss to match the segmentation output leads to poor performance on both new and old categories (-5.8% on overlapped and-6.3% on disjoint, with respect to ℓ_{LOC}). This is because the softmax normalization in the cross-entropy loss does not treat each class independently and causes the localizer to generate high scores for old categories even when they have low segmentation scores.

Smoothing effect on pseudo-supervision. We tune α, the hyper-parameter in Equation 3.4 that regulates the smoothness of the pseudo-labels used to supervise the segmentation model. Fig. 3.3 displays the final mIoU in the VOC 10-10 disjoint and overlap scenarios for five different values of α ranging from 0 to 1. As expected, using hard labels ($\alpha = 1$) leads to the worst results, as the model tends to fit the noise in the supervision and forgets prior knowledge, failing in learning novel classes. We selected $\alpha = 0.5$ for our experiment because it is a reasonable trade-off in terms of mIoU between learning and remembering. It is worth noting that changing α from 0 to 0.7 has only a minor impact on the results, with an average difference of less than 0.5% between the disjoint and overlap cases, indicating the robustness of WILSON to different values of α.

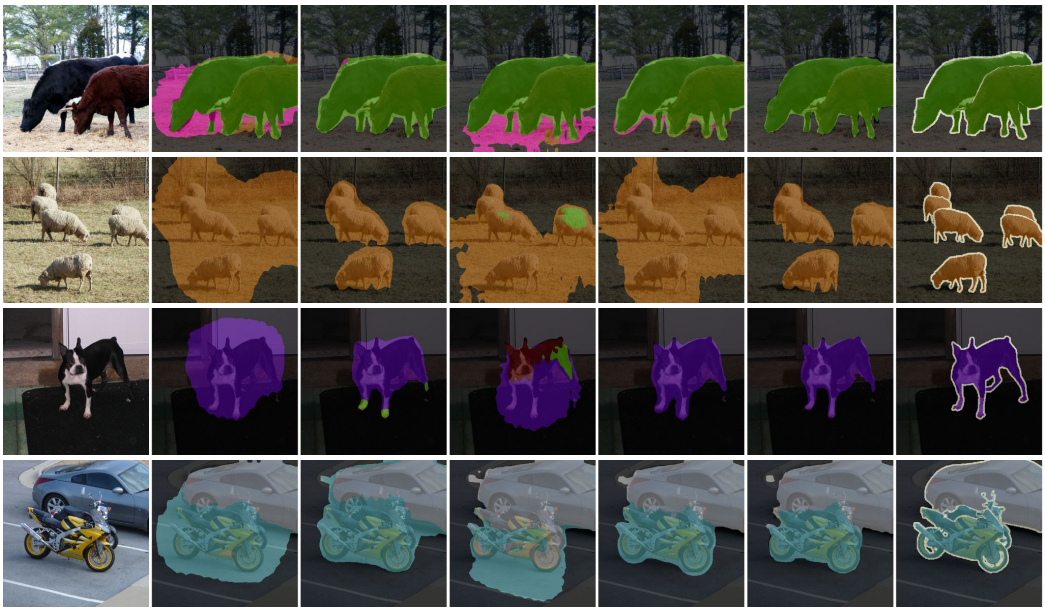


Fig. 3.4 Qualitative results on Pascal VOC 10-10 setting comparing different weakly supervised semantic segmentation approaches. The image emphasized the superiority of WILSON in both learning new classes (e.g. sheep, dog, motorbike) and preserving knowledge of old ones (e.g. cow, car) with respect to competitors. From left to right: image, CAM, SEAM [3], SS [4], EPS [5], WILSON and the ground-truth. Best viewed in color.

Using supervision for all the classes. Since image-level labels are not expensive, we evaluate weakly-supervised techniques' performance when the supervision in the incremental steps is provided for both old and new categories. The results of this evaluation on the Pascal VOC dataset are shown in Table 3.5. When comparing these results to Table 3.1 and Table 3.2, a performance improvement can be noticed. In particular, all of the methods experience improvements, with WILSON achieving an average of $+2\%$ on both old and new categories in the 15-5 and 10-10 scenarios. This demonstrates that integrating knowledge about old categories in the pseudo-supervision generation is critical for both learning new categories and at the same time avoiding forgetting of old ones. Additionally, we show that in this scenario, WILSON outperforms all the offline weakly-supervised semantic segmentation methods. Specifically, WILSON achieves better performance in every setting, outperforming EPS on the VOC 15-5 by 1.2% and 0.1%, and by 1.8% and 1.3% on the VOC 10-10, for disjoint and overlapped scenario, respectively.

Limitations. Although WILSON achieves impressive results in multi-class incremental learning using image-level labels, it lacks the capability to perform single-class

incremental learning due to the structure of Eq. (3.3). Specifically, Eq. (3.3) requires the presence of negative examples in the training batch, which poses a challenge in the context of single-class incremental learning where such examples are absent. Consequently, this restriction hampers the usability of WILSON in scenarios where the updates of the model require the addition of only one new class at a time.

3.3.5 Conclusions

In this section we presented WILSS, a new setting in which the goal is to extend semantic segmentation models' knowledge via cheap image-level annotations. Directly applying weakly-supervised semantic segmentation strategies to incremental learning methods would require generating pseudo-supervision offline, and training the segmentation model after. Instead, we proposed WILSON, that combines the segmentation model with a localizer and uses image-level labels on new categories to generate online pseudo-supervision for the segmentation backbone. We demonstrated that adding a localization prior (coming from the old model) to the localizer improves the generation of pseudo-annotations. We proved the efficacy of our approach in three incremental settings and showed that it outperforms weakly-supervised semantic segmentation methods that generate pseudo-annotations offline. Furthermore, we achieved results close to fully supervised incremental learning state-of-the art methods.

Chapter 4

Towards recognizing and learning unseen new semantic concepts

While convolutional neural networks have significantly advanced robot vision, they are often limited to closed world scenarios, where the amount of semantic categories to be recognized is pre-determined by the training set. Since it is practically unfeasible to capture all the possible real world semantic concepts into a single training set, it is crucial to break the closed world assumption, and equip our robots with the ability to act in an open world. To provide such capabilities, a robot vision system should be capable of (i) detecting previously unseen concepts and ii) expanding their knowledge over time, as new semantic classes arrive. In this chapter we show how to improve deep open world recognition algorithms by enforcing a global-to-local features clustering at class level. Specifically, we start by formalizing the open world recognition (OWR) setting in Section 4.1, and we review related work in Section 4.2. We then introduce our method, called ***B-DOC*** *(**B**oosting **D**eep **O**pen World Recognition by **C**lustering) (Section 4.3). We present a global clustering loss that enforces the model to map samples closer to the centroid of their class, and a local clustering loss that shapes the latent space such that samples of the same class get closer while pushing away neighbours from other categories. Furthermore, we propose to learn class-specific rejection thresholds with a novel loss formulation, instead of heuristically estimating a single global threshold as done in previous works. Experiments on RGB-D Object, Core50 and CIFAR-100 datasets show the effectiveness of our approach. (Section 4.3.3).*

*In Section 4.4 we take a step further, investigating the effect of discrepancies between training and test distributions (*i.e.*, domain-shift) has on OWR frameworks. In Section 4.4.1 we present the first benchmark for fairly assessing OWR algorithms performance with and without domain-shift. We use this benchmark to conduct analyses in several scenarios, showing that existing OWR methods indeed suffer from a sever performance deterioration when train and test distributions differ. Our analyses show that coupling OWR algorithms with domain generalization strategies only mitigates this degradation, indicating that the mere plug-and-play of existing domain generalization algorithms is not enough to recognise new and unknown classes in unfamiliar domains (Section 4.4.2). Our results point towards open challenges and future directions, that need to be investigated for designing robot visual systems that can function reliably under these challenging yet realistic conditions (Section 4.4.5).*

4.1 Problem statement

Traditional machine learning models are trained under the closed world assumption (CWA) which implies that the only classes the models will ever need to recognize are the ones learned in the training set. However, this is an extremely limiting perspective, as in many real world applications the set of classes of interest is not known a priori, and might change over time. Thus, we need systems to be able to recognize new classes as they appear, and handle the uncertainty and ambiguity that arises from the presence of unknown classes. The open world recognition (OWR) scenario aims at breaking the CWA, empowering models for the recognition of both known and unknown classes in a real world setting. To accomplish this, for a robot vision system is crucial to be capable of: (i) detecting unknown categories while recognizing already seen ones, and (ii) extending its internal knowledge with new categories, without forgetting the previously learned ones having no access to old training sets (avoiding therefore the *catastrophic forgetting* issue [82]).

In this context, we show how deep OWR algorithms can be enhanced by implementing a global-to-local features clustering approach at the class level (Section 4.3.2). Specifically, we propose a *global clustering* loss that compels the model to map samples closer to their class centroid, and a *local clustering* loss that shapes the latent space such that samples of the same class are brought closer together, while pushing away at the same time samples from other categories. Moreover, we introduce a novel loss formulation for learning class-specific rejection thresholds, as opposed to heuristically estimating a single global threshold estimation as done in prior works.

In Section 4.4, we take a step further and highlight a challenge that has yet to be solved, *i.e*. these algorithms always assume that the training and test images are acquired under the same conditions. We refer to this assumption as the closed domain assumption (CDA) which, as for the CWA, may be suitable for robots operating in highly restricted settings (e.g., industrial robots) but does not apply to robots operating in the wild. These robots, indeed, need visual systems that can handle various input distributions (known as domains) that can arise from different environments, illumination, and acquisition conditions. The discrepancy between the training and test distributions is referred to as *domain-shift*. In Section 4.4.1, we introduce the first benchmark to fairly investigate whether OWR methods can effectively operate under shifting visual domains. Our findings (Section 4.4.2)

highlight the need for further research in developing robot visual systems that can reliably operate under these challenging yet realistic conditions.

Problem formulation. The purpose of OWR is to create models able to (ii) recognize known classes (*i.e.*, concepts seen during training), (ii) detect unseen classes (*i.e.*, categories not present in any previous set used for training), and (iii) incrementally add new categories as new training data becomes available. More formally, let us consider $\mathcal{X}$ to be the input space (image space) and $\mathcal{K}$ to be the closed world output space (set of known classes). In particular, as our output space will evolve as we receive more data from new classes, we denote the set of categories seen after the T_{th} incremental step as $\mathcal{K}_T$, with $\mathcal{K}_0$ denoting the classes in the first training set.

We begin with an initial training set $\mathcal{T}_0 = (x_i, y_i)_{i=1^{N_0}}$, where $x_i \in \mathcal{X}$, y_i is a class label in the set $\mathcal{Y}_0$, and N_0 is the number of samples. Subsequently, at learning step T, we receive another training set $\mathcal{T}_T$ which contains a new set of categories, *i.e.* $\mathcal{Y}_T \bigcap \mathcal{Y}_t = \emptyset \; \forall t \in [0, T-1]$. Since we aim at determining if an image contains an unknown concept, we introduce u as the special unknown category, defining the final output space as $\mathcal{K}_T$ and u. Our goal, then, is to learn a function f that maps an image x to either one of the semantic categories learned up to step T ($\mathcal{K}_T = \bigcup_{t=0}^{T} \mathcal{Y}_t$) or the unknown class u, *i.e.* $f : \mathcal{X} \rightarrow \mathcal{K}_T \cup u$.

4.2 Literature review

Open world recognition. The visual learning community has investigated in the last few years the problem of life-long, open ended learning [140, 78], with approaches tackling a unifying framework for novelty detection and incremental class learning, known as open world recognition (OWR) [6, 67, 141]. While these efforts are principled and hold promise, to the best of our knowledge their effectiveness has never been tested within the robot vision scenario. Moreover, current deep approaches in this thread address mainly the incremental class learning problem, rather than the whole OWR challenge. While previous approaches addressed the incremental and continual learning challenges, acting in real open world environments necessitates both detecting previously unseen categories and incorporating them in subsequent learning stages. In order to achieve this goal, the authors of [6] introduced the open world recognition (OWR) setting as a more generic and realistic scenario for real world agents. They also extended the Nearest Class Mean (NCM) classifier [142, 141] to OWR scenario proposing the Nearest Non-Outlier (NNO) algorithm which introduces a fixed rejection threshold that estimates if a test sample belongs to the known or unknown set of classes. [140] addressed OWR challenges with the Nearest Ball Classifier and proposed a rejection threshold based on the confidence of the model's prediction. Recently, [7] extended the NNO method of [6] by adopting as the feature extractor a trainable end-to-end deep architecture, and introduced a novel dynamic strategy to update the rejection threshold. In Section 4.3.2 we propose a novel method which introduces a global to local clustering loss able to enhance the performance of an NCM-based classifier. In addition, differently from previous works, we learn a specific rejection threshold for each class, as opposed to the previously predetermined ones based on heuristic strategies.

Domain-Shift in Robot Vision. For robotic systems to operate effectively in the wild, it is crucial to develop models robust to domain-shift. To achieve this objective, various efforts have been made in robotics to perform adaptation in the presence of target data [143–148]. However, as it is almost impossible to gather data for every possible target domain in advance, researchers have explored techniques to address the domain-shift issue without using target data during training. One approach involves utilizing a stream of incoming target data to perform online adaptation, such as updating domain-specific components [149] and/or utilizing adversarial objectives [150]. However, these strategies have a slow adaptation time, which can

be problematic in scenarios that require fast adaptation (e.g., sudden illumination changes).

One way to create systems that are robust to (possibly) any target domain without requiring any target data during training or deployment is by utilizing domain generalization (DG) techniques [151]. While existing methods focused on multi-source settings [151], there has been relatively less research devoted to the case of a single source domain. In such scenarios, where it is impossible to disentangle domain-specific and semantic-specific information explicitly, one solution is to construct classifiers that are structurally more robust using part-based models [152, 153], multiple visual cues [154], regularization strategies [155], and self-supervised learning [156]. Another approach is to simulate the presence of multiple source domains through adversarial techniques [157] or data augmentation [158]. In the latter case, data augmentation can simulate increasingly difficult new domains [158] or fictitious multiple sources [159].

In this work, we focus on DG models, which can be applied to various target domains without requiring any target data. Despite some efforts in testing domain adaptation models in an open set [160, 161] scenario, we are the first to explicitly examine the domain-shift issue in the open world recognition framework, investigating the effectiveness of coupling OWR with DG strategies to address its challenges.

4.3 Open World Recognition

Standard methods for open world recognition (OWR) [6, 140, 7] involve applying non-parametric classification algorithms on top of learned metric spaces. A popular choice is extending the Nearest Class Mean (NCM) [142, 141] classification algorithm to incorporate a rejection option based on an estimated threshold. While early approaches [6, 140] utilize shallow features, it has recently been demonstrated that deep neural networks can be effectively utilized in OWR scenarios [7]. In this thesis, we follow the deep learning approach of [7] and take a step forward.

We argue that it is essential to force the deep feature extractor to appropriately cluster samples belonging to the same class, while pushing away samples from other classes. Towards this objective, we introduce a global clustering loss term that aims at keeping the features of samples within the same class closer to their class centroid. Additionally, we show how to successfully employ the soft nearest neighbor loss [162, 163] as a local clustering loss term to force pairs of samples within the same category to be closer in the learned latent space than samples from other classes.

Moreover, unlike previous works [6, 7], we do not rely on heuristic rules to estimate a global rejection threshold on model predictions. Instead, we (i) define an independent threshold for each category and (ii) explicitly learn these thresholds through the use of a margin-based loss function, which balances errors on rejecting samples from an in memory held-out set from training. We evaluate our method, called B-DOC (Boosting Deep Open World Recognition by Clustering) on the RGB-D Object dataset [8], Core50 [9], and CIFAR-100 [10] datasets, and show that combining the two complementary clustering loss terms and learning the rejection thresholds outperforms previous methods.

Contributions. The contributions presented in this section are the following: (i) we introduce two clustering loss terms that effectively locate samples within the same category in the representation space, while separating them from samples belonging to other categories; (ii) we propose an effective method for detecting unknown samples employing learned class-specific rejection thresholds; (iii) we demonstrate the superiority of B-DOC compared to the state of the art, including quantitative analysis and an in-depth ablation of the components of our model.

	NNO	DeepNNO	B-DOC
ω	fixed	updated	updated
ϕ	$\mathcal{N}(1-\frac{d(z,\mu_y)}{\tau})$	$\exp\left(-\frac{1}{2}\|\|z-\mu_y\|\|\right)$	$\frac{1}{\varphi}z-\mu_y{}^2$
ψ	$\tau \leq 0$	$\phi \leq \tau$	$\phi > \tau$
τ	fixed	updated	learned

Table 4.1 Difference among key components of OWR methods. Each approach learns a classification function f composed of the feature extractor ω, the scoring function ϕ and the final prediction function ψ. $\mathcal{N}$ is a normalization factor, while φ is the standard deviation of the features in $z = \omega(x)$ and τ is the method-specific threshold(s).

4.3.1 Preliminaries.

OWR methodologies vary in the way the function f is defined and learned. Without loss of generality, we consider f to be built on three components: a feature extractor ω ($\omega : \mathcal{X} \rightarrow \mathcal{Z}$), which maps images into a feature space $\mathcal{Z}$; a scoring function ϕ ($\phi : \mathcal{Z} \rightarrow \Re^{|\mathcal{K}_T|}$), which maps the features to class scores for known classes; and ψ ($\psi : \Re^{|\mathcal{K}_T|} \rightarrow \mathcal{K}_T \cup u$), which maps the class scores to the final prediction. We summarize in Table 4.1 how OWR algorithms have previously defined and learned ω, ϕ, and ψ, while in the following we detail each of them.

Baselines. Standard approaches address OWR by applying non-parametric classification algorithms on learned metric spaces [6, 140]. A popular algorithm is Nearest Class Mean (NCM) [142, 141] which is adopted by the Nearest Non-Outlier (NNO) [6] to compute ϕ, which is calculated using the following equation for a known class y and a sample x:

$$\phi_y^{\text{NNO}}(z) = \mathcal{N}(1 - \frac{d(z,\mu_y)}{\tau}), \tag{4.1}$$

where $z = \omega(x)$, μ_y is the class-specific centroid computed using the NCM algorithm [142], τ is a rejection threshold computed through a set of held-out validation samples, d is a distance measure and $\mathcal{N}$ is the normalization factor. The final prediction $\psi(z)$ is computed as:

$$\psi(z) = \begin{cases} u & \text{if } \phi_y^{\text{NNO}}(z) \leq 0 \; \forall y \in \mathcal{K}_T, \\ \arg\max_{y \in \mathcal{Y}_T} \phi_y^{\text{NNO}}(z) & \text{otherwise.} \end{cases} \tag{4.2}$$

Following [142], in [6] the features are projected into a metric space defined by a matrix W (*i.e.* $\omega(x) = W \cdot x$) with W being learned in the initial training step, and kept

fixed during all the subsequent learning steps. The key limitation of this approach is that when new knowledge is added to the classifier ϕ, the feature extractor ω is not updated accordingly. A solution to this issue is presented in [7] which introduces a deep extension of NNO, named DeepNNO, that uses as features extractor ω a deep architecture trained end-to-end in each incremental step, and computes the scoring function as:

$$\phi_y^{\text{DNNO}}(z) = \exp\left(-\frac{1}{2}||z-\mu_y||\right). \tag{4.3}$$

Considering $z = \omega(x)$, at step T the feature extractor ω is trained by minimizing the binary-cross entropy loss:

$$\ell_{BCE}(z_i, y_i) = \sum_{y \in \mathcal{Y}_T} 1_{y=y_i} \log(\phi_y^{\text{DNNO}}(z_i) + 1_{y \neq y_i} \log(1 - \phi_y^{\text{DNNO}}(z_i)). \tag{4.4}$$

After the training phase, the final prediction is obtained as:

$$\psi(z) = \begin{cases} u & \text{if } \phi_y^{\text{DNNO}}(z) \leq \tau \; \forall y \in \mathcal{K}_T, \\ \arg\max_{y \in \mathcal{Y}_T} \phi_y^{\text{DNNO}}(z) & \text{otherwise} \end{cases} \tag{4.5}$$

where τ is updated through a heuristic rule that adjusts the threshold based on the predictions of the network (raised whenever the model predicts true positives or negatives, and lowered whenever it predicts false positives or negatives).

Differently from [142, 6], the data representations in the latent space change along with the parameters of the backbone. As a result, it is not possible to keep class-specific centroids μ fixed, especially in an incremental learning scenario, where changes in network parameters will cause a discrepancy between old class centroids and current network activations. Such discrepancy, indeed, cannot be recovered, since previous training sets ($\mathcal{T}_i$ with $i < T$) are unavailable. To overcome this issue, DeepNNO proposed to (i) update the class centroids in an online fashion and (ii) perform rehearsal by storing samples from old categories. It also computes a distillation loss [111, 78] on the network activations by means of the network at the previous training step. In this way, it reduced the catastrophic forgetting problem by preventing deviation between the network activations and the features used to discern old categories.

While our architecture ω and the classification functions ϕ and ψ are based on the work of [7], we argue that DeepNNO has two main drawbacks. Firstly, the

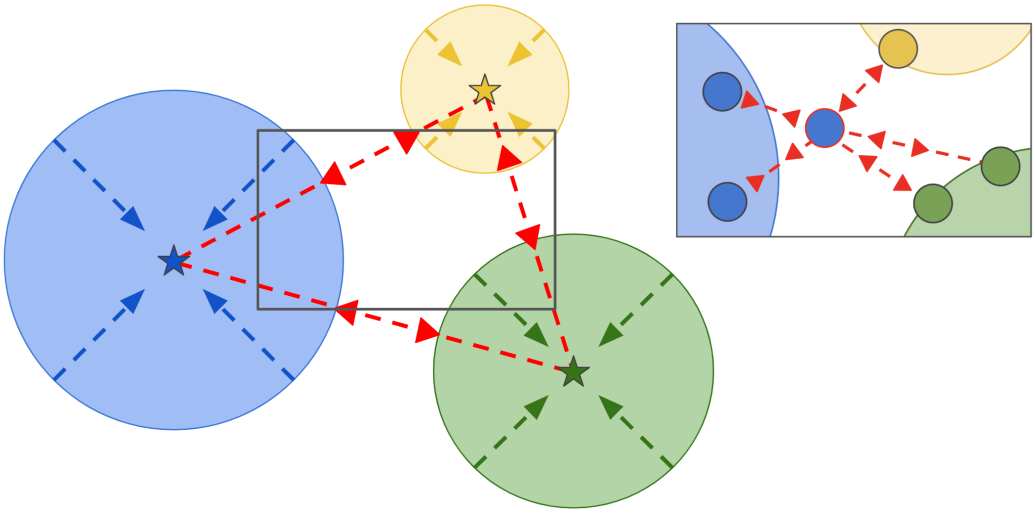


Fig. 4.1 Overview of our global to local clustering strategy. The global clustering (depicted on the left) pushes representations closer to the centroid (star) of the respective category. On the other hand, for a given sample in the representation space, the local clustering (depicted on the right forces its neighborhood to be semantically consistent, pushing away samples belonging to other categories.

learned feature representation ω is not forced to emit predictions clearly *localized* within a limited region of the metric space. Indeed, forcing the representations of a given category to a limited area of the metric space results in more confident predictions on seen categories, and clearer rejections for images of unseen classes. Secondly, its heuristic approach for setting the threshold is sub-optimal and with no guarantees on the robustness of the chosen threshold. In the following, we will present our solutions to address both these issues.

4.3.2 Boosting Deep Open World Recognition by Clustering: BDOC

In this section we introduce our core components and how we learn the rejection thresholds independently for each class. In order to enforce feature representations to be clearly localized in a limited region of the metric space based on their semantics, we introduce a pair of loss functions enforcing clustering. Specifically, we use a *global* term that enforces the network to map images of the same categories close to their respective class centroid (Fig. 4.1, left) and a *local* clustering term

that constrains the neighborhood of an image to be semantically consistent, *i.e.* to contain samples from the same category (Fig. 4.1, right). Furthermore, we introduce a distance-based loss function to learn class-specific rejection thresholds. If the distance between a sample from a certain class and its class centroid is more than the threshold, the loss forces the threshold value to be increased. On the other hand, if the distance between a sample not belonging to that class and the class centroid is less than the threshold, the loss function enforces a reduction in the threshold value.

Global Clustering. The goal of the global clustering term is to minimize the distance between the features of a sample and the centroids of its category. To model this, we utilize a cross-entropy loss with the probabilities obtained through the distances among samples and category centroids. Mathematically, let x be a sample and y its class label, the global clustering term is defined as:

$$\ell_{GC}(x,y) = -\log \frac{s_y(x)}{\sum\limits_{k \in \mathcal{K}_T} s_k(x)} \tag{4.6}$$

where the class-specific probability $s(x)$ comes from the softmaxed scores $\phi_y^{\text{BDOC}}(z)$ defined as:

$$\phi_y^{\text{BDOC}}(z) = \frac{1}{\varphi} z - \mu_y{}^2, \tag{4.7}$$

where $z = \omega(x)$, φ is set to be the standard deviation of z, to normalize the representation space and increase the system's stability.

During training, σ^2 is set to the variance of the current batch features extracted, while we also maintain an online global estimate of σ^2 that we use during testing. The class mean vectors μ_i with $i \in \mathcal{K}_T$ and σ^2 are calculated in an online fashion, as in [7].

Local Clustering. To enforce that the samples in the feature space have a semantically consistent neighborhood (*i.e.*, for a sample x of class y the nearest neighbors of $\omega(x)$ belong to category y) we employ the soft nearest neighbor loss [162, 163]. This loss measures the class-conditional entanglement of features in the representation

space and it is defined as:

$$\ell_{LC}(x,y,\mathcal{B}) = -\log \frac{\sum_{x_j \in \mathcal{B}_y \setminus \{x\}} e^{-\frac{1}{\varphi}||\omega(x)-\omega(x_j)||^2}}{\sum_{x_k \in \mathcal{B} \setminus \{x\}} e^{-\frac{1}{\varphi}||\omega(x)-\omega(x_k)||^2}} \tag{4.8}$$

where $\mathcal{B}$ represents the current training batch, $\mathcal{B}_y$ is the set of samples in the batch belonging to category y, and $\varphi = \sigma^2$.

Intuitively, for a sample x of class y, If the loss value is low, it suggests that the nearest neighbors of $\omega(x)$ belong to class y. On the other hand, if the loss value is high, it indicates that the nearest neighbors belong to classes $i \in \mathcal{K}_T$ with $i \neq y$. Minimizing the loss allows to maintain semantic consistency among the neighborhood of sample x in the representation space.

Reducing catastrophic forgetting through distillation. As highlighted in the previous sections, to avoid catastrophic forgetting, we want to preserve the behaviour learned by the feature extractor in the previous training steps. To achieve this, we follow standard rehearsal-based incremental learning approach [78, 101, 7, 80], and we introduce two elements: (i) a memory which stores the most relevant samples of the categories in $\mathcal{K}_T$, and (ii) a distillation loss that enforces consistency between the features extracted by ω and those obtained from the feature extractor of the previous learning step (*i.e.*, $\omega T - 1$).

Mathematically, the distillation loss is defined as follows:

$$\ell_{DS}(x, \omega_{T-1}) = ||\omega(x) - \omega_{T-1}(x)|| \tag{4.9}$$

This loss is minimized only during the incremental learning steps (*i.e.*, only when $T > 1$).

Overall, we train the network to minimize the following loss on a batch of samples $\mathcal{B} = \{(x_1, y_1), ..., (x_{|\mathcal{B}|}, y_{|\mathcal{B}|})\}$:

$$\mathcal{L} = \frac{1}{|\mathcal{B}|} \sum_{(x,y)\in\mathcal{B}} \ell_{GC}(x,y) + \lambda\ \ell_{LC}(x,y,\mathcal{B}) + \gamma\, \ell_{DS}(x) \tag{4.10}$$

where λ and γ are hyperparameters that weight the relative importance of each component. We set $\lambda = \gamma = 1$ in all the experiments.

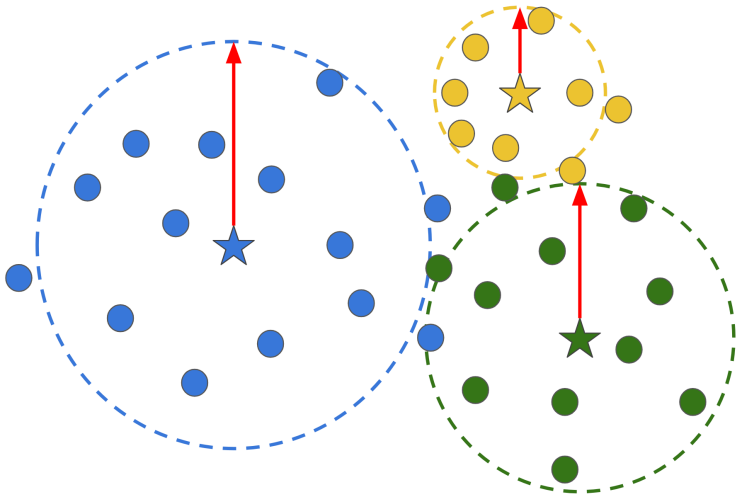


Fig. 4.2 Overview of our *class-specific* rejection thresholds learning approach. We represent the samples in the held-out set using small circles, while the centroid of each respective class using stars. The dashed circles indicate the limitis beyond which a sample is considered not a member of that class and thus rejected. The class-specific learned maximal distance used to reject a sample is depicted in red. As it can be evinced, we learn the class-specific thresholds to reduce the rejection errors.

Learning to detect the unknown. To extend our NCM-based classifier to the open set scenario we explicitly learn class-specific rejection criteria. As depicted in Fig. 4.2, we define the *class-specific* threshold τ_y for each class y as the maximum distance for which a sample still belongs to y. Under this definition, we express our ψ function as:

$$\psi(x) = \begin{cases} u & \text{if } \phi_y^{\text{BDOC}}(z) > \tau_y,\ \forall y \in \mathcal{Y}_T, \\ \text{argmin}_y \phi_y^{\text{BDOC}}(z) & \text{otherwise} \end{cases} \tag{4.11}$$

Instead of heuristically estimating or fixing a maximal distance, we explicitly learn it for each class minimizing the following objective:

$$\ell_{MD}(x,y) = \sum_{k \in \mathcal{K}_T} \max(0, m \cdot (\frac{1}{\varphi}||\omega(x) - \mu_k||^2 - \tau_k)) \tag{4.12}$$

where $\varphi = \sigma^2$, and $m = -1$ if $y = k$, $m = 1$ otherwise. If the distance between a sample of class y and its class centroid μ_y is greater than τ_y, the ℓ_{MD} loss leads to an

increase of τ_y. On the other hand, if the distance between a sample not in class y and μ_y is less than τ_y, the ℓ_{MD} loss decreases the value of τ_y.

Overall, the training procedure of B-DOC consists of two stages: in the first stage, we train the feature extractor on the training set by minimizing the loss in Eq. (4.10). In the second stage, we learn the distances τ_y using a set of samples that have been excluded from the training set. To achieve this, we split the memory samples into two parts: one is used to update the feature extractor ω and the centroids μ_y, while the other part is used for learning the τ_y values.

4.3.3 Experiments

In this section, we start by introducing the experimental setting and the metrics utilised for the evaluations. We then provide the results of our experiments followed by an ablation study on our contributions.

Datasets and Baselines. We assess the performance of B-DOC on three datasets: RGB-D Object [8] Core50 [9] and CIFAR-100 [10]. The RGB-D Object dataset [8] is widely-used to evaluate the capabilities of a model in recognizing daily-life objects (see Chapter A). It includes 51 different semantic categories, which we split into two groups for our experiments: 26 classes are considered known categories, while the other 25 are unknown classes. Among the 26 categories, we consider the first 11 as the initial training set, and then we incrementally add the remaining categories in 4 steps of 5 classes each. As in [8], we sub-sample the original dataset by taking one every fifth frame, and we follow the first train-test split among the original ones defined in [8]. In each of these splits, one object instance is selected from each class to be excluded from the training set and become part of the test set. The utilized split provides almost 35,000 training images and 7,000 test images. As done in previous approaches [6, 7], we ignored depth information to focused on RGB images only.

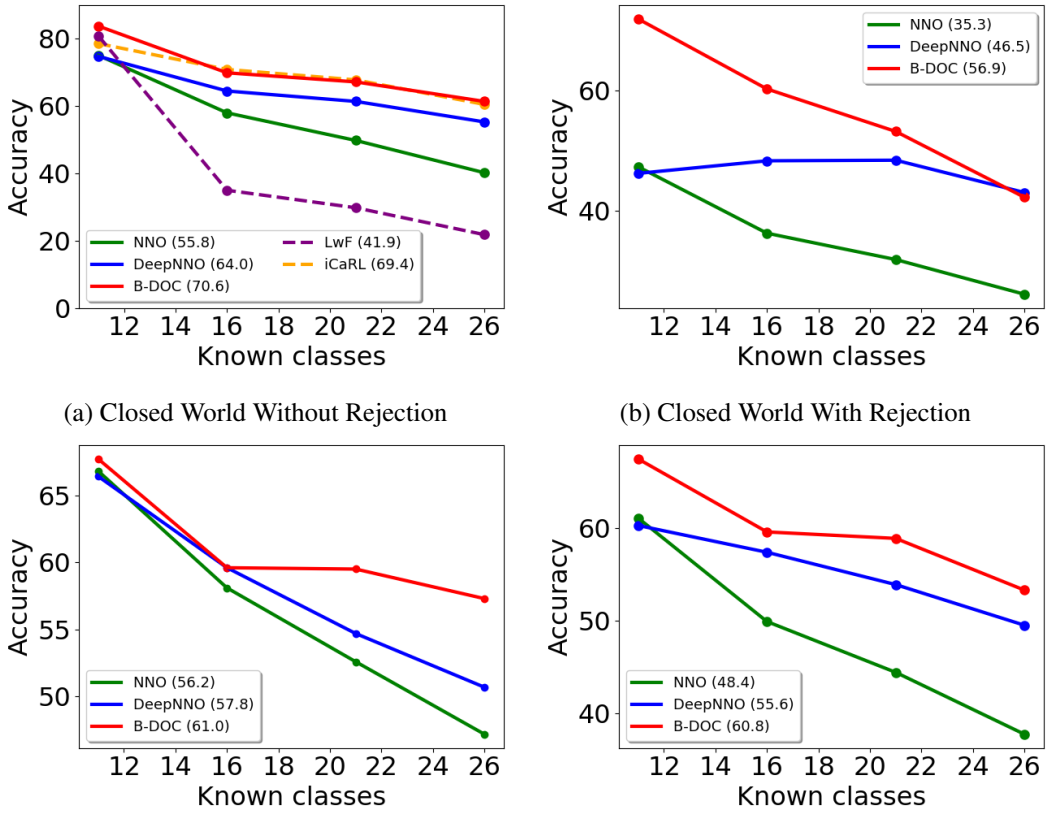
The recent Core50 benchmark [9] is mostly used to evaluate continual learning methods in an egocentric setting. It consists of images of 50 objects grouped into 10 semantic classes. Following the protocol outlined in [9], we use sequences 3, 7, and 10 for evaluation and the remaining sequences for training. Due to the differing conditions between sequences, this dataset is a particularly challenging benchmark for object recognition. Similar to RGB-D Object dataset division, we split the Core50 dataset into two parts, with 5 classes being considered as known, and the other 5 as

unknown. Within the known set, the first 2 classes are used as the initial training set. The other classes are incrementally added 1 at a time.

The CIFAR-100 dataset [10] is a standard benchmark for evaluating incremental class learning algorithms [78]. It consists of 100 semantic categories and, following previous works [7], we split them into 50 known classes and 50 unknown ones, using 20 classes as the initial training set and incrementally adding the remaining classes in steps of 10.

We compare the performance of B-DOC in the OWR scenario to DeepNNO [7] and NNO [6], using the implementation in [7] for the latter. We further compare B-DOC with two standard incremental class learning algorithms, *i.e*. LwF [77] (in the MC variant of [78]) and iCaRL [78]. Since both of them are developed for closed world scenario only, we use their results as references in that setting, without open-ended evaluation. For each dataset, we have randomly chosen five different sets of known and unknown classes and, after fixing them, we ran the experiments three times for each method. The final results are obtained by averaging the results among each run and order.

Networks architectures and training protocols. As in previous works, we employ a ResNet-18 architecture [67] for all the experiments. For both RGB-D Object dataset and Core50 we train the network on the initial classes for 12 and 4 epochs respectively, starting from scratch. For CIFAR-100 dataset, instead, we set the number of epochs to 120 for the initial learning phase and to 40 for each subsequent incremental learning step. We employ a learning rate of 0.1 and batch size equal to 128 for the RGB-D Object dataset and CIFAR-100, while 0.01 and 64 for Core50. To train the network we use Stochastic Gradient Descent (SGD) with a momentum of 0.9 and a weight decay equal to 10^{-3}. We resize the images within the RGB-D Object dataset to 64×64 pixels and those within the Core50 dataset to 128×128 pixels. We apply random cropping and mirroring to all datasets, and perform color on the set of held-out samples varying brightness, hue and saturation. For the baselines, we employ the same network architecture and training protocol outlined in [7]. We also use the same memory management strategy of [7], with a fixed size of 2000 samples, and constructing each batch by drawing 40% of the samples from memory. However, unlike in [7], we never see during training 20% of the samples present in memory, we use them only to learn the class-specific thresholds values τ_y.



(a) Closed World Without Rejection

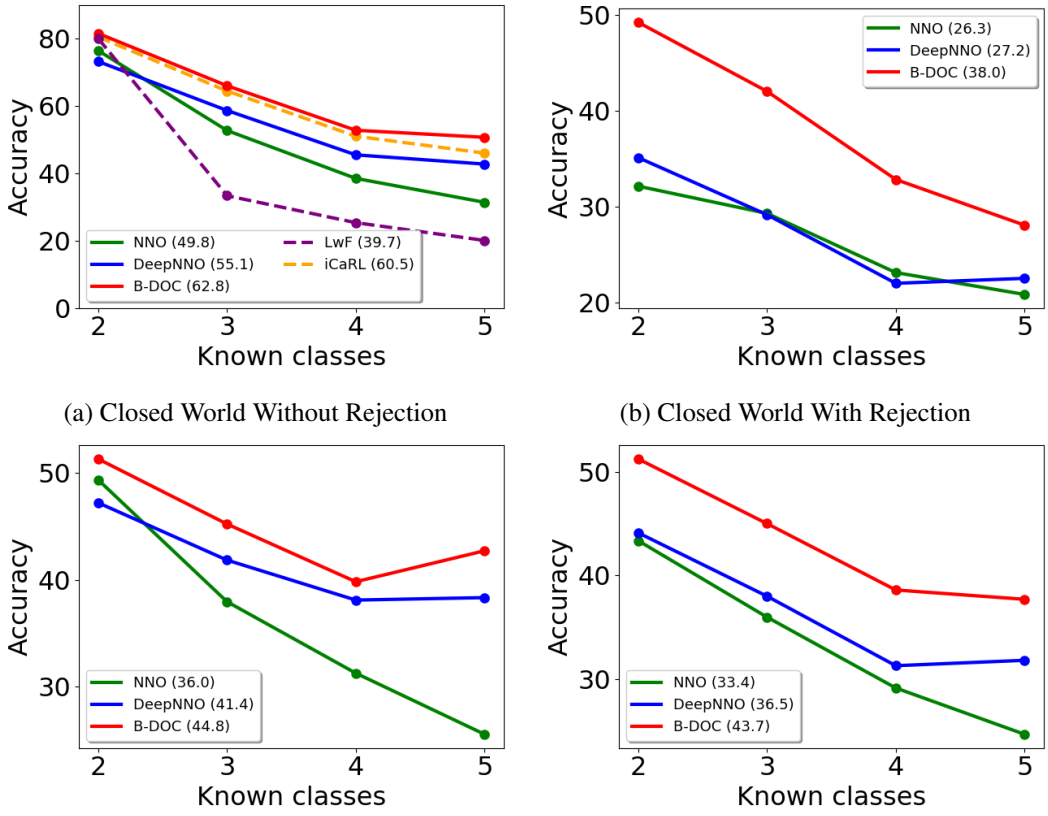
(b) Closed World With Rejection

(c) Open World Recognition Average

(d) Open World Recognition Harmonic Mean

Fig. 4.3 Comparison of NNO [6], DeepNNO [7] and B-DOC on RGB-D Object dataset [8]. The average accuracy among the different incremental steps is indicated in parenthesis.

Metrics. We evaluate the performance of OWR methods using 3 standard metrics. For the closed world, we show the global accuracy *with* and *without* rejection option. In particular, in the closed world *without rejection* scenario, the model is only tested on the known set of categories, *excluding* the possibility to identifying a sample as *unknown*. This setting measures the model's capabilities of correctly classifying samples withing the given set of categories. In the closed world *with rejection* setting, instead, the model can either categorize a samples withing the known set of categories, or classifying it as *unknown*. This scenario is more challenging than the one without rejection, because samples within the known set of categories may be misclassified as *unknowns* instead. For the open world scenario, we use the standard OWR metric defined in [6], which is the average accuracy between the accuracy achieved by the model in the closed world *with rejection* scenario and the accuracy achieved in the open set scenario (i.e., the accuracy at rejecting samples of



(a) Closed World Without Rejection

(b) Closed World With Rejection

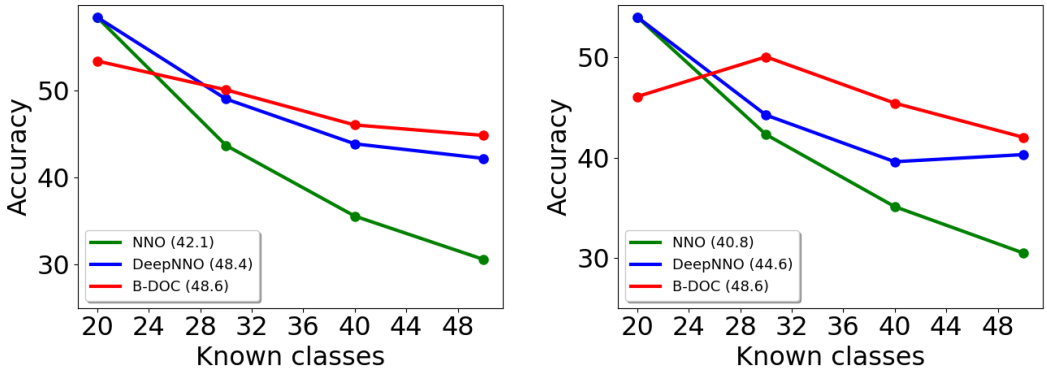
(c) Open World Recognition Average

(d) Open World Recognition Harmonic Mean

Fig. 4.4 Comparison of NNO [6], DeepNNO [7] and B-DOC on Core50 dataset [9]. The average accuracy among the different incremental steps is indicated in parenthesis.

unknown categories). However, this metric can create biases in the final score (e.g., the accuracy of a method that rejects every sample will be 50%). To mitigate this bias, we introduced the OWR-H metric, which is the harmonic mean between the accuracy in the open set and in the closed world with rejection scenarios.

Quantitative results. The results of our method on the RGB-D Object dataset are shown in Fig. 4.8. When operating in the closed world without rejection (Fig. 4.8a) scenario, B-DOC is able to improve the feature representation, therefore outperforming DeepNNO and NNO by 5.6% and 14.8% of accuracy on average, respectively. This improvement is due to the introduction of global and local clustering loss terms, which combined together allow the model to better cluster samples withing the same class and to separate them from samples of other categories. When compared to the incremental class learning approaches LwF and iCaRL, B-DOC is highly competitive, surpassing LwF by a large margin while being comparable with iCaRL. We



(a) Open World Recognition Average (b) Open World Recognition Harmonic Mean

Fig. 4.5 Comparison of NNO [6], DeepNNO [7] and B-DOC on CIFAR-100 dataset [10]. The average accuracy among the different incremental steps is indicated in parenthesis.

believe that these results are remarkable, given that our model's primary goal is not to purely incrementally extend its knowledge with new concepts. The comparison on the closed world with rejection, depicted in Fig. 4.8b demonstrates that B-DOC is also more confident in classifying known classes, being capable of rejecting a smaller number of known samples. In particular, B-DOC is more confident in the first incremental steps, achieving, on average, an accuracy that is 10.3% higher than DeepNNO. Considering open world metrics, B-DOC is superior to previous approaches. As shown in Fig. 4.3c, our method reaches similar results to DeepNNO in the first stages, but outperforms it in the latest steps. In the OWR-H metric (Fig. 4.3d), our method consistently outperforms previous methods in all incremental steps. This can be attributed to the fact that previous methods tend to reject more samples, as demonstrated by their lower closed world with rejection performance. In contrast, our learned rejection strategy, combined with our clustering losses, enables our model to achieve a better trade-off between open set and closed world with rejection accuracy. Overall, B-DOC improves on average over DeepNNO by 4.8% and 5.2% in the OWR and OWR-H metrics, respectively.

The results on the Core50 [9] dataset are shown in Fig. 4.4. Similarly to the RGB-D Object dataset, when compared to incremental learning algorithms designed for the closed world setting, B-DOC performs competitively, remarkably surpassing iCaRL by 4.7% of accuracy in the final incremental step. Moreover, our method achieves superior results with respect to OWR state-of-the-art algorithms in both the closed world with and without rejection scenarios. In the former scenario,

our method outperforms NNO by 13.01% and DeepNNO by 7.74% on average (Fig. 4.4a) and outperforms both NNO and DeepNNO by more than 10% in the latter (Fig. 4.4b). It is also worth noting that both DeepNNO and NNO struggle to model the confidence threshold, rejecting most of the samples with known classes. Indeed, by incorporating the rejection option, the accuracy drops considerably for DeepNNO and NNO, down to 27.2% and 26.3% respectively, while B-DOC achieves an average accuracy of 38.0%. We report in Fig. 4.4c and Fig. 4.4d the performances on Core50, under standard and harmonic OWR metric. consistently with previous results, B-DOC outperforms DeepNNO by 3.4% and 7.2% in average in standard OWR and OWR-H metrics respectively, confirming the efficacy of the proposed clustering losses and the learned class-specific maximal distances.

Finally, the results on the CIFAR-100 dataset are shown in Fig. 4.5a and 4.5b in terms of the OWR and OWR-H metrics, respectively. Even in this benchmark, our method performs better on average than previous approaches. While our model achieves lower performance than NNO and DeepNNO in the initial training stage, which we attribute to a poor initial estimation of the rejection thresholds, it outperforms both methods in the subsequent incremental learning steps, demonstrating the ability of our model to learn and recognize unknown new classes in an open world, without forgetting old categories. In fact, considering the incremental steps, our model shows an average improvement of 10% over NNO in both the OWR and OWR-H metrics, and an improvement of 2% and 4.5% over DeepNNO in OWR and OWR-H metrics, respectively.

Ablation studies

The main components of our approach are three: (i) the global clustering loss (GC); (ii) the local clustering loss (LC); (iii) the learned class-specific rejection thresholds. In this section, we will examine in-depth each of these contributions, starting with the clustering loss terms and then comparing the choice we made for the rejection strategy to other common options.

Global and local clustering. In Table 4.2, we compare the performance of the two clustering loss terms considering the OWR metrics in the RGB-D Object dataset. Evaluating the global clustering (GC) and the local clustering (LC) terms separately, we find that on average they show similar performance. In particular, using only

Method	Known Classes				OWR	
	11	16	21	26	[20]	H
GC	66.0	57.3	58.6	53.3	58.8	58.7
LC	64.1	56.0	57.9	56.4	58.6	58.4
Triplet	62.1	54.9	54.8	49.5	55.4	55.4
GC + LC	**67.7**	**59.6**	**59.5**	**57.3**	**61.0**	**60.8**

Table 4.2 Ablation study on three clustering approaches: global clustering (GC), local clustering (LC) and Triplet loss, under the OWR metric. The average OWR-H over all steps is shown in the right column.

Method	Class specific	Multi stage	Known	Unknown	Diff.
DeepNNO [7]			84.4	98.8	14.4
Ours	✓		83.0	98.6	15.6
		✓	4.4	26.9	22.6
	✓	✓	27.4	65.2	**37.8**

Table 4.3 Ablation study on the rejection rates of different approaches for detecting unknowns. Results computed on the RGB-D Object dataset using the same feature extractor.

GC, we achieve slightly better results on the first three incremental stages, while LC performs better on the fourth step. However, the best results on every steps are achieved by combining together global and local clustering terms (GC + LC). This demonstrates that the two clustering terms provide complementary contributions and help to learn a representation space that properly clusters samples within the same categories, while better detecting unknowns. We also report in Table 4.2 the results of using a triplet loss [164] instead of our objective function, since they both share the same learning objective, *i.e.* shaping a metric space in which samples sharing the same semantics are closer then samples with different semantics. As the table shows, the triplet loss (Triplet) achieves significantly inferior performance than our full objective function, with a gap of over 5% in both the standard and harmonic OWR metrics. Notably, it achieves lower results than all of the loss terms in isolation, and the superior performance of local clustering term further confirms the advantages of SNNL-based loss functions compared to triplets, as previously shown in [163].

Detecting the Unknowns. In Table 4.3, we provide a comparison of different approaches for rejecting samples on the RGB-D Object dataset [8], all using the same feature extractor. Specifically, we compare our method that learns class-

specific maximal distances with the three other baselines: (i) the strategy proposed by DeepNNO [7], (ii) learning class-specific maximal distances but during training (i.e. unlike our two-stage pipeline), and (iii) learning a single maximal distance that applies to all the categories using our two-stage strategy. In the comparison, we consider the difference in the rejection rates for both known and unknown samples. For known class samples, we report the percentage of the samples which have been correctly classified in the closed world but then rejected when the rejection option was included. We intentionally exclude the wrongly classified samples in order to isolate rejection mistakes from classification errors. For unknown samples, we report the open set accuracy, which corresponds to the percentage of rejected samples among all the unknown samples. The third column shows the difference between the open set accuracy and the rejection rate on known test samples. Ideally, we would like to have this difference as close as possible to 100%, indicating a 100% rejection rate on unknown samples and 0% on samples belonging to known classes.

The table shows that our two-stage pipeline with class-specific maximal distances strategy reaches the highest gap, rejecting 27.4% of known class samples and 65.2% of unknown samples. This gap we have with other strategies is remarkable. Employing the two-stage pipeline but with a class-*generic* maximal distance results in a low rejection rate for both known and unknown samples, with a difference of 22.6%, which is 15.2% lower than using a class-*specific* distance. Differently, estimating the confidence threshold as in DeepNNO [7], or without the two-stage pipeline, leads to a very high rejection rate for both known and unknown classes, resulting in a difference of 14.4% and 15.6% for DeepNNO and the single-stage strategy respectively. These results are the lowest two among the four strategies.

In fact, calculating the thresholds using only the training set introduces a bias in the rejection criterion towards the overconfidence the model acquired on this set. During testing, this causes the model to consider differences in the data distribution (such as different object instances) as a source for rejection, even if test samples contain known semantics. The two-stage strategy allows to overcome this bias and to tune the rejection criterion on unseen samples, on which the model cannot be overconfident.

4.3.4 Conclusions

In this section, we proposed a method for addressing the open world recognition problem in robot vision. Like previous approaches, our method is based on a NCM classifier built upon a deep feature extractor. We improved the OWR performance of this framework by introducing as a training objective the minimization of a global to local clustering loss. This loss helps to reduce the distances between samples within the same class in the latent space, while increasing the distances between samples of different classes, resulting therefore in a better detection of unknown concepts. Additionally, we explicit learn class-specific distances to determine when a sample should be rejected, rather than relying on heuristic estimates as prior works. We evaluated our method, B-DOC, on standard recognition benchmarks and demonstrated superior performance compared to the previous OWR state-of-the-art.

While in this section we focused on the OWR scenario, there are still many directions that would need to be explored to fully enable robots to learn autonomously in real world scenarios. In the following section, we will discuss the challenges these approaches face under shifting visual domains, and we will see that while there exist some techniques to mitigate this issues, the field remains still open.

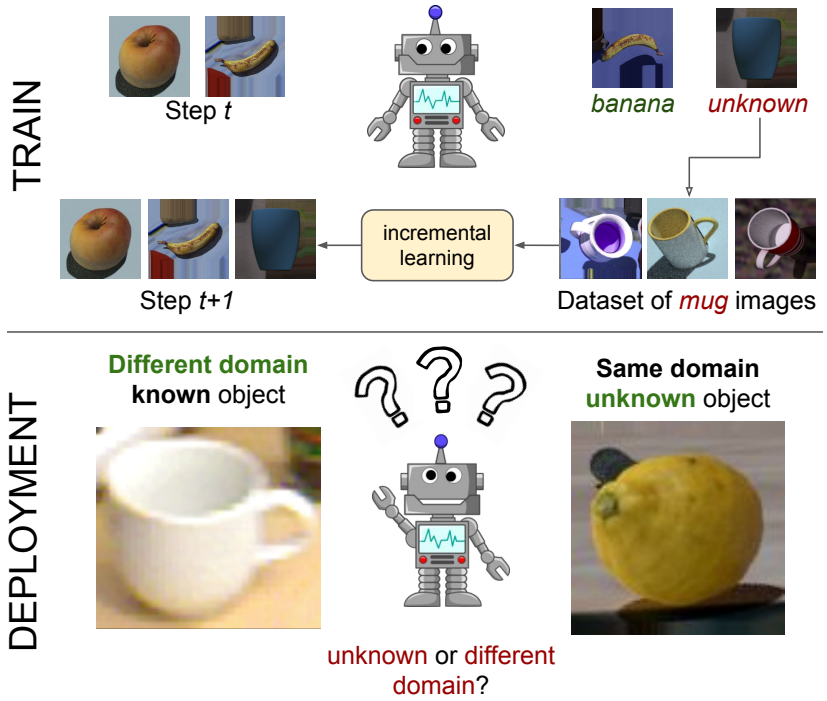


Fig. 4.6 Overview of our considered problem. In OWR, an agent must be able to incrementally learn new concepts over time while detecting previously unseen ones. Our research question is: does the efficacy of the visual system hold when operating in various visual domains and environments?

4.4 Open World Recognition under Shifting Visual Domains

Identifying the presence and the semantic category of an object in an image is a crucial ability for any visual understanding system. As presented in Section 4.1, a primary issue of traditional object recognition systems is that they often rely on the CWA, meaning that the set of classes available during training is assumed to be the only one that the robot will ever encounter in the real world. This assumption is unrealistic for robots acting in various and dynamic environments, given that there are potentially an infinite number of semantic concepts in the real world. The need of breaking the CWA has lead researchers to explore the OWR problem [6], in which algorithms are asked to not only detect previously unseen semantic concepts, but also learn new categories over time.

Although their efficacy, these algorithms overlook the possibility of operating in neverseen conditions, assuming that training and test images will always come from the same exact conditions. As for CWA, this assumption, which we refer to as the closed domain assumption (CDA), is only appropriate for robots operating in highly constrained environments such as industrial robots. However, the CDA is unrealistic for e.g., mobile robots operating in the wild, and their visual systems needs to be able to handle the various input distributions (*i.e.*, domains) that can arise from e.g., different environments, illumination, acquisition conditions. For example, a visual system for security robots patrolling public areas, that has been trained on purely daytime images, may struggle to generalize to nighttime images due to the significant differences between the two. This difference between the training and test distributions is known as domain-shift [165]. Several works have addressed this problem in robot vision in the form of domain adaptation (DA) [145, 11]. In standard domain adaptation settings, we have labeled data for one training domain (*source* domain) and unlabeled data for one test domain (*target* domain), and the objective is to utilize these data to model the discrepancies between the source and target distributions.

Since our ultimate goal is to break both the CWA and CDA (see Fig. 4.6) simultaneously, a crucial yet still open question is whether OWR algorithms can properly function under domain-shift. In this section, we attempt to answer this question by benchmarking OWR approaches under variations between training and test distributions. To fulfill our objective, we evaluate the three OWR algorithms introduced in Section 4.3.1, *i.e.* NNO [6, 140], DeepNNO [7] and B-DOC, on the widely-adopted RGB-D Object dataset (ROD) [8] and two additional datasets sharing the same semantic classes, but different acquisition conditions, namely synthetic ROD (synROD) [11] and Autonomous Robot Indoor Dataset (ARID) [12]. When trained on either synROD or ROD and tested on the other datasets, we observe for OWR algorithms significant performance degradation, with drops down to almost 45% in the OWR harmonic mean, demonstrating how OWR approaches severely suffer from performance deterioration when tested on domains on which they have not been trained on. Interestingly, we found that end-to-end trained deep OWR algorithms are more susceptible to the domain-shift problem than their non end-to-end counterparts, despite achieving the highest performance on in-domain tests.

In the next sections, we then combine NNO, DeepNNO, and B-DOC with three single source domain generalization (DG) techniques. The results of our

experiments showed that DG techniques mitigate but do not completely solve these issues, highlighting how the goal of solving CWA and CDA jointly is still far from being accomplished.

Contributions. To summarize, the contributions presented in this section are the following: (i) we present the first benchmark of OWR algorithms under shifting domains, showing that their performance significantly decreases when tested onto different domains; (ii) we demonstrate that combining OWR models with single-source domain generalization techniques only partially solves the problem, but does not eliminate it; (iii) we propose a validation methodology for allowing fair and easy future research.

4.4.1 Shifting visual domains benchmark

In this section, we describe the single source domain generalization algorithms we use in our benchmark. We refer the reader to Section 4.1 for the problem notation.

Data augmentation with transformation sets (RSDA). The first common single source domain generalization approach is via data augmentation techniques, either adversarial [157, 159] or transformations-based [158]. In this section, as representative of this category, we select the data augmentation based approach of [158]. Given a training batch $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{n}$, we train the model applying semantic objectives on a transformed version of the same batch $\hat{\mathcal{B}} = \{(\alpha x_i, y_i)\}_{i=1}^{n}$, where α is a randomly sampled transformation from the set $\mathcal{A}$. The set $\mathcal{A}$ populated by composed transformations that are extracted from a set of simple transformations A (e.g., blurring, mirroring). In particular, an evolutionary-based algorithm selects the combination of A that results in the worst model performance and adds it to $\mathcal{A}$. The hyperparameters of the model include the set A of basic transformations, their possible values, and the frequency of updating $\mathcal{A}$. We fix the set A to the following transformations: hue, contrast, brightness, saturation, random crop, and mirroring.

Self-supervised learning with relative rotations (RR). Another widely used approach to improving domain generalization performance is through self-supervised learning [156]. In particular, by employing an auxiliary self-supervised task, the model concentrates on discriminative invariances and regularities, thereby enhancing generalization to new domains [156]. To be effective, the task must require the model to focus on the content of images, rather than their peculiar styles or appearances.

Solving jigsaw puzzles [166, 156] and predicting rotations [167, 161] are examples of successful tasks. Here, we take the relative rotations [161] approach. In particular, given a batch $\mathcal{B}$, we create a new batch $\hat{\mathcal{B}} = \{(x_i, y_i, \text{rot}_{\theta_i}(x_i), \theta_i)\}_{i=1}^{n}$ where rot represents a rotation transformation applied with angle θ_i to the original image x_i. θ_i is sampled from a discrete set Θ (*i.e.*, 0°, 90°, 180° or 270°), making the auxiliary task classifying which θ_i has been applied to x_i. To perform this, we instantiate a network branch ρ that maps features extracted from both the original and rotated images to the correct rotation angle, i.e. $\rho : \mathcal{Z} \times \mathcal{Z} \rightarrow \Theta$. We use a standard cross-entropy loss on the rotation predictions to update both ρ and ω. In the training objective of an OWR algorithm (e.g. Eq.(4.10)), this auxiliary loss is included scaled by a parameter ξ. Note that we apply the semantic objectives also to the rotated images and we apply random data augmentations to them, in order to increase the complexity of the auxiliary task.

Regularization through self-challenging (SC). Finally, regularization techniques can enhance the ablities of models to generalize to unseen domains [155]. Here we evaluate the self-challenging algorithm proposed in [155], where the model is asked to classify features that have been corrupted by removing the elements that mostly contributed to a correct classification of the corresponding sample. Formally, features ($z = \omega(x)$) are extracted from the original samples to calculate a score $\phi_y(z)$ for the ground-truth class y. The gradient of the score with respect to z is then computed ($g = \partial \phi_y(z)/\partial z$), and a new set of features ($\hat{z} = m \circ z$) is obtained by applying a binary mask m on the original features z, where $\circ$ is the Hadamard product. The mask m has 0s for values z_j whose gradients $g_j \geq q_p$, and 1s for others. The threshold q_p is determined by preserving the top-p percentile of activations per corrupted sample. The model's hyperparameters are the sample- and batch-wise corruption ratios.

Experimental setting. We perform the experiments on three datasets: RGB-D Object dataset (ROD) [8], synthetic ROD (synROD) [11], and Autonomous Robot Indoor Dataset (ARID) [12]. All three datasets contain images of the same 51 daily-life objects but obtained through varying acquisition conditions.

In **ROD** [8], objects are captured in a controlled scenario, with no clutter or changes in illumination or background, only varying camera angles. On the other hand, **ARID** [12] presents a more challenging environment, with objects depicted against various backgrounds, scales, views, lighting, and occlusions. It was designed

to test the robustness of recognition models in real world conditions. **synROD** [11] is a synthetic version of ROD [8] aimed at evaluating a model's ability to handle the domain shift between synthetic and real images. For more detailed information, see Chapter A.

Concerning the evaluation protocol, we adopt the same division presented in Section Section 4.3.3, using 26 classes as known and 25 as unknown, and incrementally adding 5 classes in each step with the first 11 serving as base classes. We used the first train-test split defined in [8], including one instance per class in the test set and the rest in the training set. For synROD, we follow the split outlined in [11], adding the images of the 3 classes omitted from the benchmark[1]. Finally, we use ARID [12] exclusively for testing, being the most challenging and realistic scenario.

Metrics. To evaluate the performance of OWR methods under domain-shifts, we use two metrics: i) the accuracy of known classes (closed world with and without the rejection) across all the incremental steps to assess the capability of learning new concepts and ii) the open world harmonic mean (OWR-H) to assess OWR performance (see Section 4.3.3 for additional information).

Validation protocol. In OWR, an open question is how to select the values of a method's hyperparameters since i) in each training step only a subset of the semantic categories is available, and ii) samples of unknown classes are not available. Here, we present a strategy that uses only the base categories to find the optimal hyperparameters.

To solve this, we propose to split the categories available in the first training stage into two groups: known and unknown. 10% of the base classes (e.g. 2 out of 11 in our benchmark) are considered unknown categories and the rest are considered known. From the known class set, 50% of them (e.g. 5 out of 9 in our benchmark) are used for the initial training step, while the other half (4 out of 9 in our case) are used for the incremental learning steps. Note that with this setup we artificially created i) a set of base classes to initiate the training of the model, ii) a set of categories that will be incrementally learned, and iii) a set of unseen categories to evaluate the model's open set performance. Finally, since the number of categories received in each incremental step is unknown during deployment, we simulate this uncertainty using multiple trials with different cardinality in terms of incrementally

[1]The omitted images were provided by the authors of [11].

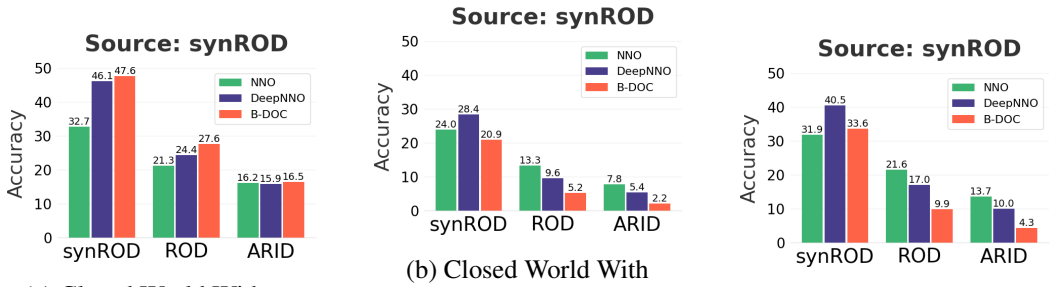
added classes. In particular, we use multiple steps with a single class (4 steps with 1 class in our benchmark), two steps with half of the classes (2 steps of 2 classes), and a single step with all the classes (1 step with 4 classes). The known/unknown and base/incremental splits are repeated several times to improve hyperparameter value estimation.

Once obtaining the base class splits, we perform hyperparameters validation in two steps. Firstly, we validate hyperparameters that contribute *only* to build the network's knowledge (i.e., closed world without rejection). This ensures that the model can effectively learn new concepts without the risk of retaining only the old knowledge, or having low confidence on its prediction, which could negatively impact open set performance later. In this stage, two types of hyperparameters are validated: those related to network optimization (e.g., learning rate and weight decay) and those related to the weights of the loss function (e.g., cross-entropy, distillation, and clustering). In the second step, all hyperparameters related to detecting samples containing unknown classes are validated using OWR-H performance. For example, we validate the negative weight used by DeepNNO [7] to update the rejection thresholds, and the learning rate used by B-DOC to learn the class-specific rejection thresholds.

Note that this entire process is agnostic to the OWR model and benchmark adopted, relying solely on the base classes to determine the optimal hyperparameters. The only hyperparameters this protocol does not set are the training epochs for the base and incremental steps, which we determine by evaluating the training accuracy on the categories present in each learning step. In our case, we used 12 epochs for ROD and 70 for synROD for the base categories. For the incremental steps, we use a proportionate number of epochs for ROD based on the number of added classes, while we have fixed the value to 35 for synROD, as it is a more challenging scenario.

4.4.2 Experiments

In this section, we present the results of our benchmark. Specifically, we evaluate the performance of standard OWR methods under domain shift (Section 4.4.3), considering both Synthetic-to-Real and Constrained-to-Unconstrained scenarios, showing that OWR algorithms suffer from significant performance degradation whenever their input distributions change. Next, we demonstrate that using single



(a) Closed World Without Rejection

(b) Closed World With Rejection

(c) OWR Harmonic mean

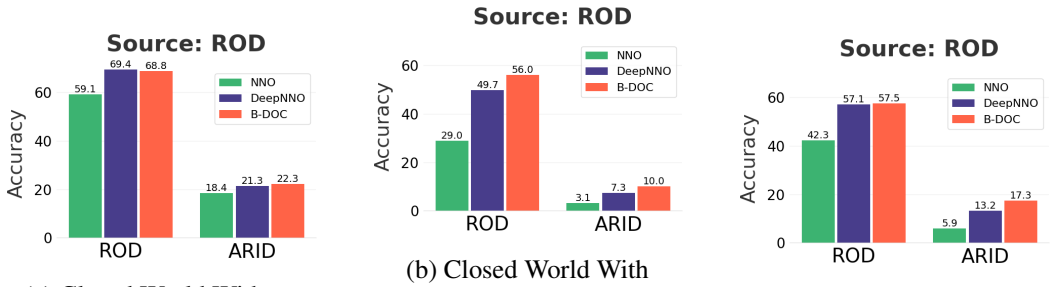
Fig. 4.7 Comparison of NNO [6], DeepNNO [7] and B-DOC (Section 4.3) trained on synROD [11] and evaluated on synROD [11], ROD [8] and ARID [12]. Numerical values denote the average accuracy among the different incremental steps.

source domain generalization (DG) techniques in combination with OWR methods can mitigate the domain-shift issue, despite being still not sufficient to fully solve the problem (Section 4.4.4). Finally, we discuss the implications of our benchmark, highlight open issues, and state future research directions (Section 4.4.5).

4.4.3 Are OWR models Robust to Domain Shift?

Synthetic-to-Real. We start our analysis by considering the synROD dataset as the source domain, and all the other datasets are target domains, in turn. The results are presented in Fig. 4.7 in terms of closed world with and without rejection, and OWR harmonic mean. As shown, all OWR methods experience a significant decline in performance when evaluated under domain-shift. In particular, we can see in Fig. 4.7a that in the closed world setting *without* the possibility of categorizing samples as unknowns, recognizing real objects is a very challenging task for all OWR algorithms trained on synthetic data. While DeepNNO and B-DOC achieve good performance in absence of domain-shift (47.6% and 46.1%, respectively), they experience a drop in performance by almost 18% when going from synROD to ROD, and by almost 26% when going from synROD to the more challenging ARID.

Similarly, in Fig. 4.7b, we see that the performance with the rejection option enabled decreases in average by almost 15% on ROD, and by more than 19% on ARID. Surprisingly, the performance of B-DOC is the most affected when the rejection option is introduced, resulting in a loss of nearly 16% accuracy on ROD and 20% on ARID. This may be because the thresholds used by B-DOC are estimated



(a) Closed World Without Rejection

(b) Closed World With Rejection

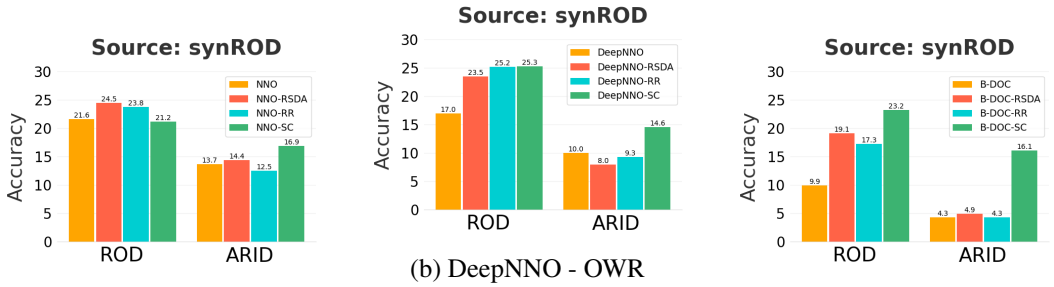
(c) OWR Harmonic mean

Fig. 4.8 Comparison of NNO [6], DeepNNO [7] and B-DOC (Section 4.3) trained on ROD [8] and tested on ROD [8] and ARID [12]. Numerical values denote the average accuracy among the different incremental steps.

using a holdout set from the training data, which do not accurately represent the distribution of test samples under domain-shifts. As a result, the wrongly computed thresholds lead the model to poor performance. Similarly, DeepNNO performs poorly in the closed world with rejection scenario, achieving values of 9.6% accuracy on ROD and 5.4% on ARID. Surprisingly, NNO performs considerably better on ROD and ARID, with an average loss of 13.5%. The reason of this is that NNO usually computes a low threshold on synROD, due to both the low confidence its shallow classification model has in the predictions, and to the high variability of the dataset. This enables NNO to reject fewer samples and better preserve closed world accuracy. However, the average accuracy of nearly 5% on ARID raises serious concerns about the applicability in real scenarios of these algorithms

Lastly, we can consider Fig. 4.7c as a global analysis, from which it is clear that the OWR-H performance of all methods confirms previous trends, suffering a significant drop, with an average decrease of more than 19% on ROD and 26% on ARID. In particular, the performance of both deep models (*i.e.* DeepNNO, B-DOC) experiences a decrease of almost 23% on ROD and 30% on ARID. Again (and surprisingly), NNO exhibits a good trade-off, with a performance decrease of around 15% on average.

Constrained-to-Unconstrained. We continue our set of experiments in a different, real-to-real scenario, where the source domain is ROD and the target domains are ROD and ARID. We note that both ROD and ARID are real datasets, which differ only from the environments they depict: constrained (ROD) and unconstrained (ARID).



(a) NNO - OWR Harmonic (b) DeepNNO - OWR Harmonic (c) B-DOC - OWR Harmonic

Fig. 4.9 Comparison of NNO [6], DeepNNO [7] and B-DOC (Section 4.3) coupled with Domain Generalization techniques when trained on synROD [11] and evaluated on ROD [8] and ARID [12]. Numerical values denote the average accuracy among the different incremental steps.

The results in Fig. 4.8 show that, as expected, while all the methods achieve good performance on the same domain, they all suffer a significant decrease in performance when tested under domain-shifts. However, this decline is even more pronounced than the one seen in the synthetic-to-real scenario. In the closed world with rejection scenario (Fig. 4.8a), the models which had an average accuracy of 65% on ROD, experience a drop of almost 45% in accuracy when tested on ARID, down to almost 20% of accuracy. The same drop in accuracy is observed in the closed world with rejection setting (Fig. 4.8b). The domain-shift causes the models to confuse samples from the new domain as unknowns, as evidenced by the drop in performance between ROD and ARID: the accuracy with rejection, in the latter, is barely of 7% on average.

Overall, the performance under the OWR-H metric is very unsatisfactory, showing a significant gap (almost 40% on average) between the accuracy values obtained on ROD and ARID by all three methods. These results highlight how OWR methods significantly suffer when tested on data coming from new domains/environments and, confirms that domain-shift is a major issue for OWR algorithms.

4.4.4 Can DG methods address the problem?

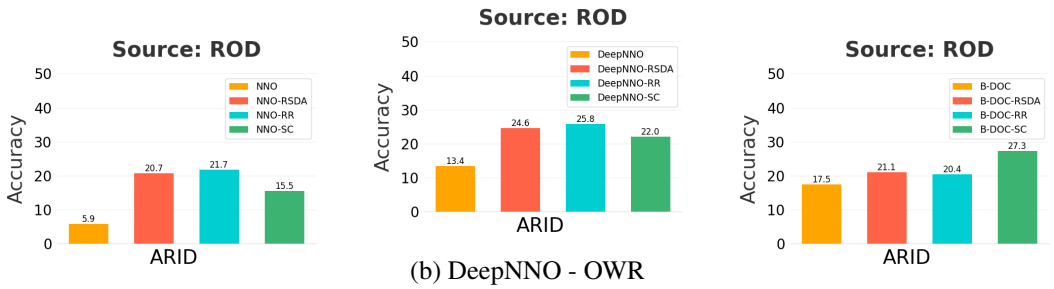
Given the unsatisfactory performance of OWR methods under domain-shift, in the following section we investigate whether single source domain generalization algorithms can be used to address and solve this problem.

Synthetic-to-Real. We start our analysis from the synthetic-to-real scenario. Fig. 4.9 shows how OWR-H changes when OWR methods are coupled with DG techniques. While NNO experiences a slight improvement, the performance of both DeepNNO and B-DOC gains a large boost, particularly when the SC strategy [155] is applied. In fact, with respect to the originals, the results on ARID for DeepNNO and B-DOC are 2 and 4 times higher, respectively. Considering the other DG methods, they all contribute to improving DeepNNO and B-DOC performance on ROD, although with zero to minor benefits on ARID. We attribute the higher efficacy of SC technique to its regularization of the classifier, which forces it to i) focus on several cues and ii) achieve a lower confidence on the predictions, which may leads to estimate better thresholds for identifying unknowns in different domains. As stated above, all the domain generalization techniques have little to even negative impact on the performance of NNO (Fig. 4.9a). The reason for this is that NNO does not fine-tune its representation across training steps, and this limits the impact DG techniques might have on learning a more domain invariant latent space.

For what concerns the other domain generalization techniques, while RR [161] brings a small improvement across all baselines in each scenario, RSDA [158] is more effective on ROD dataset, rather than ARID. The reason behind it is that synROD images differ mainly in color and shape from ROD ones, and using specific data augmentation to bridge these differences results in a global performance improvement. For ARID, the impact of RSDA is only partial due to other (different) challenges such as occlusion and scale variations.

Despite these results, the domain-shift problem remains still significantly present, with an average drop in performance from the synROD results of approximately 10% on ROD and 19% on ARID.

Constrained-to-Unconstrained. Lastly, in this section we examine the impact of DG techniques on models trained on ROD and evaluated on ARID dataset. The results are reported in Fig. 4.10. As shown in the figure, all of the DG methods improve OWR methods in this scenario, albeit in different ways. SC, for example, leads to the best results for B-DOC, with 27.3% accuracy, while RSDA and RR are more effective for DeepNNO and NNO. In particular, even if the training phase in which the DG algorithms operate is limited, NNO still benefits from their adoption in this scenario, achieving results that are more than three times greater when combined



(a) NNO - OWR Harmonic (b) DeepNNO - OWR Harmonic (c) B-DOC - OWR Harmonic

Fig. 4.10 Comparison of NNO [6], DeepNNO [7] and B-DOC [13] with Domain Generalization techniques when trained on ROD [11] and tested on ARID [12]. The numbers denote the average accuracy among the different incremental steps.

with RR and RSDA. This implies that the augmentations applied by these methods aids in reducing the domain-shift between ROD and ARID.

However, when compared to the original results on ROD dataset, the gaps remain significant: NNO+RR achieves still 40% less in accuracy than the original NNO on ROD. Similarly, DeepNNO+RR and B-DOC+SC are almost 30% lower than their respective counterparts evaluated on ROD. These results confirm that the domain-shift issue in OWR can only be mitigated (but not solved) by coupling OWR methods with single-source DG algorithms.

4.4.5 Conclusions

Based on the findings in Section 4.4.2, two key conclusions can be drawn. Firstly, OWR methods lack robustness to domain-shift, resulting in significantly poorer performance when evaluated on data from different distributions than the ones seen during training. Secondly, despite minor improvements, combining OWR models with single-source DG techniques is not sufficient to fully solve the domain-shift issue. Following these conclusions, it is worth emphasizing two open issues that we believe require further attention in future research, in order to develop OWR systems suitable for real world use cases.

Domain-shift in recognition. In order to effectively apply OWR models in real world scenarios, they need to adapt to unseen domains quickly. In Section 4.4.4 we have shown how coupling OWR methods with DG techniques is not sufficient for addressing this issue, resulting in poor recognition performance and incorrect

estimation of rejection thresholds. Possible future works include developing a single model to address both problems incorporating algorithms that utilize target data streams in an online DA fashion [149].

Domain-shift while learning. Existing OWR methods assume that data in each incremental step comes from the same distribution. However, an open question is how the performance of these methods would be affected if the data in different incremental steps were from different domains. Our intuition is that models would be more prone to forgetting, and would also tend to use domain cues to make (incorrect) predictions. A crucial challenge in this scenario would be disentangling domain- and semantic-specific. Specifically, in the incremental learning step, it becomes vital to guide the model's attention towards visual cues that are representative of the semantic of the novel class, while disregarding those influenced by the current domain. By doing so, the model could improve its ability to effectively generalize and accurately identify the novel category across diverse domains. Possible directions may either involve the usage of unlabeled data [168] or additional side-information [169].

Chapter 5

Conclusions

This chapter summarizes key contributions and findings presented in this thesis, and further discusses open issues and potential directions of future research.

5.1 Summary of contributions

In this thesis, we investigated the ability of deep neural networks to recognize previously unseen semantic concepts and incorporate knowledge that was not part of the original training set, with the goal of developing deep models that are capable of recognizing and learning new/unseen visual semantic concepts over time.

In Chapter 2, we started by introducing the main limitation on which traditional machine learning models rely, *i.e*. the closed world assumption (CWA). Under this assumption, models confine their understanding of the world to a specific set of categories they have been trained on, overlooking the inevitable possibility of having to deal with unexpected, unseen categories once deployed in the real world. After a literature review (Section 2.2), we formally introduced the anomaly segmentation (AS) scenario in Section 2.3.1, and analyzed the drawbacks of the popular approach called MSP, which relies on the highest probability assigned to any of the known classes to compute the anomaly score for a pixel. To overcome the softmax function limitations of MSP, we argued that anomaly scores should be computed directly from the classification scores, and we introduced **PAnS** (Section 2.3.2) that learns class-specific prototypes through a cosine classifier and computes the anomaly scores based on the classifier predictions. Section 2.3.3 reported the experimental results which supported the effectiveness of our method.

As a second step towards developing models able to act in the real world, we introduced the incremental learning (IL) scenario in Section 3.1. We provided an extensive review of related work in Section 3.2, and we then introduced a more challenging yet realistic scenario, where pre-trained deep models are asked to extend their knowledge using only cheap image-level labels (Section 3.3). After the problem mathematical formulation (Section 3.3), we introduced our method **WILSON**, that combines the segmentation network with a localizer module, and leverages image-level labels on new categories to generate pseudo-labels for the segmentation model in one single step. Section 3.3.2 and 3.3.3 detailed the components of WILSON, and Section 3.3.4 presented our qualitative and quantitative results, showing how WILSON is capable of outperforming weakly-supervised semantic segmentation approaches, and achieving comparable results to standard fully supervised IL methods.

Finally, in Section 4.1 we presented the open world recognition (OWR) framework, which seeks to break the CWA by allowing models to detect the presence of never-seen-before categories and learn new ones as they become available. After a review of OWR strategies in Section 4.2, we presented our methods **B-DOC** (Section 4.3), which introduces a global-to-clustering training loss objective, and a learnable rejection threshold per each class in order to distinguish between known and unknown categories. Our experimental results and findings were presented in Section 4.3.3. Moreover, in Section 4.4 we took a step further, and we investigated the effects that visual domain-shifts have on OWR methods. We presented the first benchmark to fairly assess OWR methods under shifting visual domains in Section 4.4.1, and we discussed our experimental findings in Section 4.4.2. In particular, we investigated how domain-shift affects OWR methods in Section 4.4.3, and whether domain generalization approaches can alleviate the degradation in performance in Section 4.4.4.

5.2 Open problems and future directions

While in this thesis we explored the development of deep learning models capable of identifying categories that were not previously encountered (Chapter 2), learning new semantic concepts (Chapter 3), and performing both tasks (Chapter 4) in a single fashion, multiple challenges and directions still need to be explored to enable visual systems to function effectively in real world settings.

Starting from PAnS, our proposed solution to address anomaly segmentation challenges (Section 2.3.2), a major open problem remains the model's uncertainty on pixels that lie on the boundaries between different classes. It would be interesting to investigate how to strengthen the confidence of the model on those pixels. One direction could be applying conditional random fields [170] to get smoother and more coherent predictions on boundaries. Moreover, it would be interesting to analyze how transformer-based architectures [37, 38, 40] act in an anomaly segmentation scenario. Given their ability to incorporate semantic context, and the outstanding results achieved over traditional convolutional-based networks in semantic segmentation [171–175], it would be interesting to understand if they are intrinsically able to reduce the amount of pixels wrongly identified as anomalous at object boundaries, being them able to capture finer contextual relations.

For what concerns incremental learning in semantic segmentation (Section 3.3), one interesting direction would be investigating how to make WILSON able to perform single-class incremental learning steps, since to properly guide the training, Eq. (3.3) requires negative examples in the training batch. Given that WILSON still needs a considerable amount of images to be trained, another interesting research question would be how to learn novel categories using only a limited amount of images (as few-shot [176–178] approaches aims at doing). Finally, the current version of WILSON does not specifically address the background-shift issue identified by [44], which manifests itself in traditional fully-supervised semantic segmentation models, and it would be interesting to verify the effectiveness of [44]'s approach in this scenario as well.

Moving to the open world recognition task (Section 4.3), a clear limitation of OWR approaches regards the need of collecting and labelling a new set of images every time a new class is discovered, usually involving a human in the loop. While the first issue might be addressed by acquiring images from the Web [7], the second

challenge remains still unsolved. One approach could be adopting an active learning-based pipeline [179, 180] to identify the most meaningful data, and to avoid a human-in-the-loop, it could be interesting to investigate how to generate pseudo-labels to avoid the labelling process.

References

[1] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2337–2346, 2019.

[2] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. A benchmark for anomaly segmentation. *arXiv preprint arXiv:1911.11132*, 2019.

[3] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12275–12284, 2020.

[4] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4253–4262, 2020.

[5] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5495–5505, 2021.

[6] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *CVPR-15*.

[7] Massimiliano Mancini, Hakan Karaoguz, Elisa Ricci, Patric Jensfelt, and Barbara Caputo. Knowledge is never enough: Towards web aided deep open world recognition. In *ICRA-19*.

[8] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *ICRA-11*.

[9] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *CoRL-17*.

[10] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Technical report, University of Toronto., 2009.

[11] M. R. Loghmani, L. Robbiano, M. Planamente, K. Park, B. Caputo, and M. Vincze. Unsupervised domain adaptation through inter-modal rotation for rgb-d object recognition. *RA-L*, 5(4):6631–6638, 2020.

[12] Mohammad Reza Loghmani, Barbara Caputo, and Markus Vincze. Recognizing objects in-the-wild: Where do we stand? In *ICRA-18*.

[13] Dario Fontanel, Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Boosting deep open world recognition by clustering. *IEEE Robotics and Automation Letters*, 5(4):5985–5992, 2020.

[14] Andrea Maracani, Umberto Michieli, Marco Toldo, and Pietro Zanuttigh. Recall: Replay-based continual learning in semantic segmentation. In *Int. Conf. Comput. Vis.*, pages 7026–7035, 2021.

[15] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Plop: Learning without forgetting for continual semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4040–4050, 2021.

[16] Dario Fontanel, Fabio Cermelli, Massimiliano Mancini, and Barbara Caputo. Detecting anomalies in semantic segmentation with prototypes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–121, 2021.

[17] Fabio Cermelli, Dario Fontanel, Antonio Tavera, Marco Ciccone, and Barbara Caputo. Incremental learning in semantic segmentation from image labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4371–4381, 2022.

[18] Dario Fontanel, Fabio Cermelli, Massimiliano Mancini, and Barbara Caputo. On the challenges of open world recognition under shifting visual domains. *IEEE Robotics and Automation Letters*, 6(2):604–611, 2020.

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[20] 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 Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2017.

[21] Matthias Haselmann, Dieter P Gruber, and Paul Tabatabai. Anomaly detection using deep learning based image completion. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1237–1242. IEEE, 2018.

[22] Yingda Xia, Yi Zhang, Fengze Liu, Wei Shen, and Alan Yuille. Synthesize then compare: Detecting failures and anomalies for semantic segmentation. *arXiv preprint arXiv:2003.08440*, 2020.

[23] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[24] Krzysztof Lis, Krishna Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the unexpected via image resynthesis. In *Int. Conf. Comput. Vis.*, pages 2152–2161, 2019.

[25] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Eur. Conf. Comput. Vis.*, 2018.

[26] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[27] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[28] Zhenli Zhang, Xiangyu Zhang, Chao Peng, Xiangyang Xue, and Jian Sun. Exfuse: Enhancing feature fusion for semantic segmentation. In *Eur. Conf. Comput. Vis.*, 2018.

[29] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, 2017.

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

[31] Changqian Yu, Jingbo Wang, Changxin Gao, Gang Yu, Chunhua Shen, and Nong Sang. Context prior for scene segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12416–12425, 2020.

[32] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. *Eur. Conf. Comput. Vis.*, 2020.

[33] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1857–1866, 2018.

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

[35] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3146–3154, 2019.

[36] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7151–7160, 2018.

[37] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7262–7272, 2021.

[38] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[39] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021.

[40] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022.

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

[42] Umberto Michieli and Pietro Zanuttigh. Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1114–1124, 2021.

[43] Umberto Michieli and Pietro Zanuttigh. Knowledge distillation for incremental learning in semantic segmentation. *Computer Vision and Image Understanding*, 205:103167, 2021.

[44] 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.

[45] Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. In *Adv. Neural Inform. Process. Syst.*, pages 2902–2913, 2019.

[46] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. pages 1050–1059, 2016.

[47] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Adv. Neural Inform. Process. Syst.*, pages 5574–5584, 2017.

[48] Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. In *Adv. Neural Inform. Process. Syst.*, pages 5541–5552, 2018.

[49] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[50] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020.

[51] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Adv. Neural Inform. Process. Syst.*, pages 7167–7177, 2018.

[52] Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation. In *Proceedings of the ieee/cvf international conference on computer vision*, pages 5128–5137, 2021.

[53] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

[54] Giancarlo Di Biase, Hermann Blum, Roland Siegwart, and Cesar Cadena. Pixel-wise anomaly detection in complex driving scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16918–16927, 2021.

[55] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *International MICCAI Brainlesion Workshop*, pages 161–169. Springer, 2018.

[56] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[57] Matej Grcić, Petra Bevandić, and Siniša Šegvić. Densehybrid: Hybrid anomaly detection for dense open-set recognition. In *European Conference on Computer Vision*, pages 500–517. Springer, 2022.

[58] Samantha Guerriero, Barbara Caputo, and Thomas Mensink. Deepncm: Deep nearest class mean classifiers. In *ICLR-WS*, 2018.

[59] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4367–4375, 2018.

[60] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5822–5830, 2018.

[61] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3474–3482, 2018.

[62] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Adv. Neural Inform. Process. Syst.*, pages 4077–4087, 2017.

[63] Chunjie Luo, Jianfeng Zhan, Xiaohe Xue, Lei Wang, Rui Ren, and Qiang Yang. Cosine normalization: Using cosine similarity instead of dot product in neural networks. In *International Conference on Artificial Neural Networks*, pages 382–391. Springer, 2018.

[64] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 3630–3638, 2016.

[65] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019.

[66] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10951–10960, 2020.

[67] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016.

[68] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[69] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.*, 88:303–338, 2009.

[70] 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 *Eur. Conf. Comput. Vis.*, pages 740–755. Springer, 2014.

[71] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[72] 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.

[73] 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.

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

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

[76] 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.

[77] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2935–2947, 2017.

[78] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2001–2010, 2017.

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

[80] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Eur. Conf. Comput. Vis.*, pages 233–248, 2018.

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

[82] 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.

[83] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *Eur. Conf. Comput. Vis.*, pages 549–565. Springer, 2016.

[84] Bowen Cheng, Omkar Parkhi, and Alexander Kirillov. Pointly-supervised instance segmentation. *arXiv preprint arXiv:2104.06404*, 2021.

[85] Paul Vernaza and Manmohan Chandraker. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7158–7166, 2017.

[86] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3159–3167, 2016.

[87] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, pages 1635–1643, 2015.

[88] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 876–885, 2017.

[89] Pedro O Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1713–1721, 2015.

[90] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*, pages 1796–1804, 2015.

[91] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *Eur. Conf. Comput. Vis.*, pages 695–711. Springer, 2016.

[92] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 248–255, 2009.

[93] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4981–4990, 2018.

[94] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *Eur. Conf. Comput. Vis.* Springer, 2020.

[95] A. Pronobis, L. Jie, and B. Caputo. The more you learn, the less you store: Memory-controlled incremental svm for visual place recognition. *Image and Vision Computing*, 28(7):1080–1097, 2010.

[96] Gert Cauwenberghs and Tomaso Poggio. Incremental and decremental support vector machine learning. In *Advances in neural information processing systems*, pages 409–415, 2001.

[97] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

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

[99] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[100] 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.

[101] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547, 2018.

[102] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5138–5146, 2019.

[103] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *arXiv preprint arXiv:1705.08690*, 2017.

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

[105] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11321–11329, 2019.

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

[107] Enrico Fini, Stéphane Lathuiliere, Enver Sangineto, Moin Nabi, and Elisa Ricci. Online continual learning under extreme memory constraints. In *Eur. Conf. Comput. Vis.*, pages 720–735. Springer, 2020.

[108] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Eur. Conf. Comput. Vis.*, pages 86–102. Springer, 2020.

[109] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. *arXiv preprint arXiv:2111.11326*, 2021.

[110] 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.

[111] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. 2015.

[112] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019.

[113] Fabio Cermelli, Antonino Geraci, Dario Fontanel, and Barbara Caputo. Modeling missing annotations for incremental learning in object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3700–3710, 2022.

[114] 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.

[115] Akshita Gupta, Sanath Narayan, KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Ow-detr: Open-world detection transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9235–9244, 2022.

[116] Dongbao Yang, Yu Zhou, Aoting Zhang, Xurui Sun, Dayan Wu, Weiping Wang, and Qixiang Ye. Multi-view correlation distillation for incremental object detection. *Pattern Recognition*, 131:108863, 2022.

[117] 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. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 113–126, 2019.

[118] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE international conference on computer vision*, pages 3400–3409, 2017.

[119] KJ Joseph, 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*, 44(12):9209–9216, 2021.

[120] Can Peng, Kun Zhao, and Brian C Lovell. Faster ilod: Incremental learning for object detectors based on faster rcnn. *Pattern Recognition Letters*, 140:109–115, 2020.

[121] 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.

[122] 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 *ICCV*, pages 1742–1750, 2015.

[123] Meng Tang, Abdelaziz Djelouah, Federico Perazzi, Yuri Boykov, and Christopher Schroers. Normalized cut loss for weakly-supervised cnn segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1818–1827, 2018.

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

[125] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5267–5276, 2019.

[126] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7014–7023, 2018.

[127] Seong Joon Oh, Rodrigo Benenson, Anna Khoreva, Zeynep Akata, Mario Fritz, and Bernt Schiele. Exploiting saliency for object segmentation from image level labels. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5038–5047. IEEE, 2017.

[128] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2209–2218, 2019.

[129] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8991–9000, 2020.

[130] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2921–2929, 2016.

[131] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1568–1576, 2017.

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

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

[134] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2623–2632, 2021.

[135] Yi Li, Zhanghui Kuang, Liyang Liu, Yimin Chen, and Wayne Zhang. Pseudo-mask matters in weakly-supervised semantic segmentation. In *Int. Conf. Comput. Vis.*, pages 6964–6973, 2021.

[136] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? pages 6448–6458. PMLR, 2020.

[137] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1209–1218, 2018.

[138] 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.

[139] 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.

[140] Rocco De Rosa, Thomas Mensink, and Barbara Caputo. Online open world recognition. *arXiv:1604.02275*, 2016.

[141] Samantha Guerriero, Barbara Caputo, and Thomas Mensink. Deep nearest class mean classifiers. In *ICLR-WS*, 2018.

[142] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *ECCV-12*.

[143] Markus Wulfmeier, Alex Bewley, and Ingmar Posner. Addressing appearance change in outdoor robotics with adversarial domain adaptation. In *IROS-17*.

[144] Kuan Fang, Yunfei Bai, Stefan Hinterstoisser, Silvio Savarese, and Mrinal Kalakrishnan. Multi-task domain adaptation for deep learning of instance grasping from simulation. In *ICRA-18*.

[145] Gabriele Angeletti, Barbara Caputo, and Tatiana Tommasi. Adaptive deep learning through visual domain localization. In *ICRA-18*.

[146] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *CVPR-19*.

[147] Rae Jeong, Yusuf Aytar, David Khosid, Yuxiang Zhou, Jackie Kay, Thomas Lampe, Konstantinos Bousmalis, and Francesco Nori. Self-supervised sim-to-real adaptation for visual robotic manipulation. In *ICRA-20*.

[148] Jingwei Zhang, Lei Tai, Peng Yun, Yufeng Xiong, Ming Liu, Joschka Boedecker, and Wolfram Burgard. Vr-goggles for robots: Real-to-sim domain adaptation for visual control. *RA-L-19*, 4(2):1148–1155.

[149] Massimiliano Mancini, Hakan Karaoguz, Elisa Ricci, Patric Jensfelt, and Barbara Caputo. Kitting in the wild through online domain adaptation. *IROS-18*.

[150] Markus Wulfmeier, Alex Bewley, and Ingmar Posner. Incremental adversarial domain adaptation for continually changing environments. In *ICRA-18*.

[151] Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. Robust place categorization with deep domain generalization. *RA-L-18*, 3(3).

[152] Peter Uršič, Aleš Leonardis, Matej Kristan, et al. Part-based room categorization for household service robots. In *ICRA-16*, 2016.

[153] Massimiliano Mancini, Samuel Rota Bulò, Elisa Ricci, and Barbara Caputo. Learning deep nbnn representations for robust place categorization. *RA-L*, 2(3):1794–1801, 2017.

[154] Anwesan Pal, Carlos Nieto-Granda, and Henrik I Christensen. Deduce: Diverse scene detection methods in unseen challenging environments. In *IROS-19*, pages 4198–4204. IEEE, 2019.

[155] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *ECCV-20*.

[156] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR-19*.

[157] Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *CVPR-18*.

[158] Riccardo Volpi and Vittorio Murino. Addressing model vulnerability to distributional shifts over image transformation sets. In *ICCV-19*.

[159] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *CVPR-20*.

[160] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *ICCV-17*.

[161] Silvia Bucci, Mohammad Reza Loghmani, and Tatiana Tommasi. On the effectiveness of image rotation for open set domain adaptation. In *ECCV-20*.

[162] Ruslan Salakhutdinov and Geoff Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *Artificial Intelligence and Statistics*, 2007.

[163] Nicholas Frosst, Nicolas Papernot, and Geoffrey E. Hinton. Analyzing and improving representations with the soft nearest neighbor loss. In *ICML-19*.

[164] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks.

[165] Gabriela Csurka. A comprehensive survey on domain adaptation for visual applications. In *Domain adaptation in computer vision applications*, pages 1–35. Springer, 2017.

[166] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV-16*.

[167] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR-18*.

[168] Jogendra Nath Kundu, Rahul Mysore Venkatesh, Naveen Venkat, Ambareesh Revanur, and R Venkatesh Babu. Class-incremental domain adaptation. In *ECCV-20*.

[169] Massimiliano Mancini, Zeynep Akata, Elisa Ricci, and Barbara Caputo. Towards recognizing unseen categories in unseen domains. In *ECCV-20*.

[170] John D. Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, 2001.

[171] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. *arXiv preprint arXiv:2204.07143*, 2022.

[172] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[173] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. *Advances in neural information processing systems*, 34:18590–18602, 2021.

[174] Jiaqi Gu, Hyoukjun Kwon, Dilin Wang, Wei Ye, Meng Li, Yu-Hsin Chen, Liangzhen Lai, Vikas Chandra, and David Z Pan. Multi-scale high-resolution vision transformer for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12094–12103, 2022.

[175] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. *arXiv preprint arXiv:2105.05633*, 2021.

[176] Qi Fan, Wenjie Pei, Yu-Wing Tai, and Chi-Keung Tang. Self-support few-shot semantic segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIX*, pages 701–719. Springer, 2022.

[177] Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. Adaptive prototype learning and allocation for few-shot segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8334–8343, 2021.

[178] Zhihe Lu, Sen He, Xiatian Zhu, Li Zhang, Yi-Zhe Song, and Tao Xiang. Simpler is better: Few-shot semantic segmentation with classifier weight transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8741–8750, 2021.

[179] S Alireza Golestaneh and Kris M Kitani. Importance of self-consistency in active learning for semantic segmentation. *arXiv preprint arXiv:2008.01860*, 2020.

[180] Alexander Vezhnevets, Joachim M. Buhmann, and Vittorio Ferrari. Active learning for semantic segmentation with expected change. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3162–3169, 2012.

[181] A Dosovitskiy, G Ros, F Codevilla, A Lopez, and V Koltun. Carla: An open urban driving simulator. arxiv 2017. *arXiv preprint arXiv:1711.03938*.

[182] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *Int. Conf. Comput. Vis.*, pages 991–998. IEEE, 2011.

Appendix A

Datasets

The following section presents the datasets used in this thesis. We first present the semantic segmentation datasets used for anomaly segmentation and incremental learning, and then describe the datasets used to assess OWR challenges.

StreetHazards is a synthetic dataset proposed in the CAOS benchmark [2] to assess anomaly segmentation approaches. This database contains 5125 training images with the corresponding semantic labels, 1031 validation images with no anomalies, and 1500 test images containing anomalies. The images were generated using the Unreal Engine and the CARLA simulator [181], and the dataset includes different towns for the training, validation, and testing splits. The test images contain randomly selected anomalies from a set of 250 objects, which are placed in the images to create plausible road scenarios.

Pascal VOC 2012 [69] is a dataset containing real-life images used to assess several tasks, such as semantic segmentation and object detection. Following standard protocols [91, 93] we augmented the dataset with images from [182], reaching a total of 10582 training images and 1449 validation images. The dataset contains annotated images for 20 common object categories: aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train, television. The images in the dataset were collected from various sources, including the web, and cover a wide range of visual scenes.

MS-COCO [70] is a dataset used to evaluate large-scale models in a variety of tasks. It contains over 160k high resolution images. Each image is annotated with a set of

labels for a total of 80 annotated classes, 20 of which overlap with the annotations of Pascal VOC 2012.

RGB-D Object dataset [8] is a widely-used benchmark to evaluate models' capabilities in recognizing daily-life objects. The dataset consists of more than 40k images, organised into 300 instances and 51 classes of common indoor objects (e.g. scissors, cereal box, keyboard etc). Each object instance has been captured from three different viewpoints.

synROD [11] is a synthetic version of RGB-D Object dataset [8], generated utilizing freely available 3D models from public catalogs. To produce realistic lighting, the authors employed a ray-tracing engine in Blender to render the scenes. The objective of proposing this benchmark is to evaluate a model's capacity to cope with the domain-shift existing between synthetic and real images.

ARID [12] contains instead images captured in a much more realistic context than the previous two, despite containing the sema exact categories. In ARID, the objects are represented with several backgrounds, scales, views, lighting conditions, and different levels of occlusions. These additional challenges make ARID a more demanding dataset, originally collected to evaluate the robustness of deep recognition models in unconstrained settings.

Core50 [9] dataset is a recently adopted benchmark for evaluating continual learning algorithms in an egocentric setting. The dataset includes images of 50 objects, grouped into the following 10 semantic categories: clothing, food, household, kitchen, office, personal care, sports, tools, toys, vehicles. The images were acquired on 11 different sequences, under varying conditions.

CIFAR-100 [10] dataset includes 60,000 low-resolution images, usually split into 50,000 training images and 10,000 test images. Each of them falls under one of the 100 fine-grained annotated categories, which might be further divided into 20 coarse-grained classes.