Ph.D. DISSERTATION

Label-Efficient Learning for Object Recognition

객체 인식의 레이블 효율적 학습

BY

LEE JUNGBEOM

FEBRUARY 2023

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

Label-Efficient Learning for Object Recognition

객체 인식의 레이블 효율적 학습

BY

LEE JUNGBEOM

FEBRUARY 2023

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Label-Efficient Learning for Object Recognition

객체 인식의 레이블 효율적 학습

지도교수 윤 성 로

이 논문을 공학박사 학위논문으로 제출함

2023년 2월

서울대학교 대학원

전기 정보 공학부

이 정 범

이정범의 공학박사 학위 논문을 인준함

2023년 2월

위 원 장: 이 경 무

부위원장: 윤 성 로

위 원: 한 보 형

위 원: 조 남 익

위 원: 윤 상 두

Abstract

Advances in deep neural network approaches have produced tremendous progress in object recognition tasks, but it has come at the cost of annotating a huge amount of training images with explicit localization cues. To use object recognition tasks in real-life applications requires a large variety of object classes and a great deal of labeled data for each class. However, labeling pixel-level annotations of each object class is laborious, and hampers the expansion of object classes. The need for such expensive annotations is sidestepped by weakly supervised learning, in which a DNN is trained on images with some form of abbreviated annotation that is cheaper than explicit localization cues. In the dissertation, we study the methods of using various form of weak supervision, *i.e.,* image-level class labels, out-of-distribution data, and bounding box labels.

We first study image-level class labels for weakly supervised semantic segmentation. Most of the weakly supervised methods on image-level class labels depend on attribution maps from a trained classifier, but their focus tends to be restricted to a small discriminative region of the target object. We theoretically discuss the root cause of this problem, and propose three novel techniques to address this issue. However, built on class labels only, the produced localization maps are known to suffer from the confusion between foreground and background cues, *i.e.,* spurious correlation. We address the spurious correlation problem by utilizing out-of-distribution data. Finally, methods based on class labels cannot separate different instance objects of the same class, which is essential for instance segmentation. Therefore, we utilize bounding box labels for weakly supervised instance segmentation as boxes provide information about individual objects and their locations.

Experimental results show that annotation cost for learning semantic segmentation and instance segmentation can be significantly reduced: On the challenging Pascal VOC dataset, we have achieved 89% of the performance of the fully supervised equivalent by using only class labels, which reduces the label cost by 91%. In addition, we have achieved 96% of the performance of the fully supervised equivalent by using bounding box labels, which reduces the label cost by 83%. We expect that the methods introduced in this dissertation will be helpful for applying deep learning based object recognition tasks in a variety of domains and scenarios.

keywords: Deep Learning, Computer Vision, Machine Learning, Weakly Supervised Learning, Semi-Supervised Learning, Semantic Segmentation, Instance Segmentation, Object Detection
student number: 2017-29653

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Starting with the success of the ImageNet classification [3], the development of deep learning has produced tremendous progress and received a lot of attention. As carefully curated datasets are publicly released and computation is accelerated by the advancement of hardware such as general purpose graphics processing unit (GPGPU), deep learning has gained unprecedented popularity, and the performance of deep neural network surpasses that of humans. Accordingly, deep learning has been applied in numerous fields: low-level vision tasks such as deblurring [4, 5] and super-resolution [6, 7], generative models [8, 9, 10], and object recognition [11, 12]. In this dissertation, we focus on object recognition tasks.

Object recognition is one of the most important and interesting tasks in the computer vision society. Object recognition is a study for identifying objects in images or videos, which includes image classification [13, 14], object detection [15, 16, 17], semantic segmentation [18, 19], and instance segmentation [20]. Deep learning has significantly improved the performance of object recognition tasks. However, behind the success of deep learning lies a massive amount of data. In order to train a deep neural network, a large-scale dataset containing manually annotated labels is essential. However, manual labeling is notoriously laborious, which hampers the object recognition tasks to be utilized in real-world applications. This problem is particularly severe in object localization tasks such as object detection and segmentation.

To learn to localize the target object in an image, exact localization cues of the target object are necessary. For example, learning object detection requires bounding box labels that fit the extent of the target object, and learning semantic segmentation requires pixel-level segmentation masks of the target object. However, obtaining such localization labels is very expensive: pixel-level annotation of images containing an average of 2.8 objects takes about four minutes [21] per image, and a single large (2048×1024) image depicting a complicated scene requires more than 90 minutes for pixel-level annotation [22]. To use semantic image segmentation in real-world applications requires a large variety of object classes and a great deal of labeled data for each class, but the cost of labels limits the expansion of object classes and the number of data for each object class.

The need for pixel-level annotation can be addressed by weakly supervised learning, in which a segmentation network is trained on images with less comprehensive annotations that are cheaper to obtain than pixel-level labels. Weakly supervised methods can use scribbles [23], points [21], bounding boxes [24, 25, 26], and class labels [27, 28, 29, 30, 31] as annotations. These forms of weak supervision can effectively reduce the annotation cost. However, learning object localization from weak supervision is not trivial because those weak labels provide limited information about object locations. Therefore, the main goal of weakly supervised learning is to compute accurate pixel-level localization from the limited information and obtain the performance as close as possible to that of the fully supervised method.

In this dissertation, we first study to utilize image-level class labels as weak supervision in Chapter 3. Labeling an image with class labels takes about 20 seconds [21], making class labels the cheapest option. In addition, many public datasets are already annotated with class labels [3, 32], and automated web searches can also provide images with class labels [33, 34, 35] although the accuracy of such labels may be low.

The most popular option to localize the object using class labels is attribution maps obtained from a trained classifier [36, 37]. Such a map identifies the image regions on which the classifier has concentrated to predict the class of the given image. However, these important, or discriminative, regions are relatively small, and most attribution maps do not represent the whole region occupied by a target object, which makes those attribution maps unsuitable for training a semantic segmentation network. Therefore, the main goal of weakly supervised semantic segmentation using class labels is to obtain properly expanded localization maps to cover more complete regions of the target object.

In Chapter 3, we introduce two pieces of research for weakly supervised semantic segmentation using class labels. First, we propose a method for extending the discriminative regions to the extent of the target object in the following research:

- Jungbeom Lee, Eunji Kim, and Sungroh Yoon. ”Anti-Adversarially Manipulated Attributions for Weakly and Semi-Supervised Semantic Segmentation.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

The proposed method is based on adversarial attack [38, 39], but with a benign purpose. Adversarial attack finds a small perturbation of an image that pushes it across the decision boundary to change the classification result. By contrast, our method operates in an anti-adversarial manner, which is the reversal of adversarial attack. It aims to find a perturbation that pushes the manipulated image away from the decision boundary. This manipulation is realized by adversarial climbing, in which an image is perturbed along pixel gradients which increase the classification score of the target class. The result is that non-discriminative regions, which are nevertheless relevant to that class, gradually become involved in the classification, so that the CAM of the manipulated image identifies more regions of the object. This technique can be applied to not only weakly supervised semantic segmentation, but also weakly supervised object localization and semi-supervised semantic segmentation.

We then analyze the root cause why the classifier produces localization maps identifying only small regions of a target object in the following research:

- Jungbeom Lee, Jooyoung Choi, Jisoo Mok, and Sungroh Yoon. ”Reducing Information Bottleneck for Weakly Supervised Semantic Segmentation.” Advances in Neural Information Processing Systems 34 (2021): 27408-27421.

We interpret the phenomenon using the information bottleneck principle [40, 41, 42, 43]. The information bottleneck theory analyzes the information flow through sequential deep neural network layers: information regarding the input is compressed as much as possible as it passes through the layers of a deep neural network, while preserving as much of the task-relevant information as possible. This is advantageous for obtaining optimal representations for classification [44, 45] but is disadvantageous when applying the attribution maps from the resulting classifier to weakly supervised semantic segmentation. The information bottleneck prevents the non-discriminative information of the target object from being considered in the classification logit, and thus, the attribution maps focus on only the small discriminative regions of the target object. We argue that the information bottleneck becomes prominent in the final layer of the deep neural network due to the use of the double-sided saturating activation function therein (*e.g.,* sigmoid, softmax). We propose a method to reduce this information bottleneck in the final layer of the deep neural network by retraining the deep neural network without the last activation function.

We also have studied the stochastic inference technique to obtain improved localization maps in the following research, but we will briefly discuss this study for conciseness.

- Jungbeom Lee, Eunji Kim, Jangho Lee, Sungmin Lee, and Sungroh Yoon. ”Ficklenet: Weakly and Semi-Supervised Semantic Image Segmentation using Stochastic Inference.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

We have achieved 89% of the performance of the fully supervised equivalent by using only class labels, which reduces the label cost by 91%. However, there is a spurious correlation problem that cannot be solved with class labels alone. In Chapter 4, we address the spurious correlation problem by using cheap external data in the following research:

- Jungbeom Lee, Seong Joon Oh, Sangdoo Yun, Junsuk Choe, and Sungroh Yoon. ”Weakly Supervised Semantic Segmentation using Out-of-Distribution Data.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

Built on class labels only, the produced localization maps are known to suffer from the confusion between foreground and background cues, *i.e.,* spurious correlation. For example, given a database of training images where trains are typically together with the railroad, a classifier erroneously assigns high localization scores on regions containing railroad [46, 47, 48, 49, 50, 51] for the class ‘train’. The same goes for frequently co-occurring foreground-background pairs like between woodpecker and tree, snowmobile and snow, and duck and water. This is a fundamental problem that cannot be solved solely with the class labels; additional information is needed to learn to fully distinguish the foreground and background cues [46, 47, 50]. Researchers have thus sought various sources of additional guidance to separate the foreground and background cues, such as image saliency [50, 52], superpixels [53, 54], and optical flows [34, 33]. However, they tend to provide inaccurate guidance of the object locations. We propose another source of data that provides a distinction between the foreground and background cues. We propose to use the out-of-distribution (OoD) data that do not contain any of the foreground classes of interest. In addition, we propose W-OoD, a metric-learning based method of training a classifier by utilizing the OoDs to separate foreground anc background cues: increase the distance between the in-distribution and OoD samples in the feature space. This forces the background cues shared by the in-distribution and OoD samples to be excluded from the feature-space representation. By introducing very little effort to collect OoD samples, we have achieved 91% of the performance of the fully supervised equivalent.

In Chapter 5, we propose a weakly supervised learning method using bounding box labels in the following research:

- Jungbeom Lee, Jihun Yi, Chaehun Shin, and Sungroh Yoon. ”BBAM: Bounding Box Attribution Map for Weakly Supervised Semantic and Instance Segmentation.” Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

The class labels are effective in learning semantic segmentation, but performance improvement has been converged due to the inherent limitations of class labels: the absence of information about locations in the image. Moreover, class labels provide no help in separating different objects of the same class, which is the goal of instance segmentation. Bounding boxes provide information about individual objects and their locations. In addition, bounding box annotation takes about 38.1 seconds per image [55], which is still attractive. Most previous works use bounding box annotations as a search space in which a class-agnostic object mask can be found by an off-the-shelf object mask generator, such as GrabCut [56] or MCG [57]. Those mask generators operate on the low-level information of images, such as the color or brightness of pixels, and this limits the quality of the resulting mask. We propose a Bounding Box Attribution Map (BBAM), which is the pixel-level method of localizing a target object inside its bounding box using a trained object detector. We make use of attribution maps obtained from the trained object detector, which highlight the image regions that the detector focuses on in conducting object detection. More specifically, we seek the smallest areas of the image from which the object detector produces almost the same result as it does from the whole image. We can utilize higher-level information from the behavior of a trained object detector by drawing on the rich semantics learned by the object detector, resulting in better performance than previous methods depending on the low-level information only.

The remaining chapters of the dissertation are organized as follows. Chapter 2 provides background and preliminaries for object recognition and label-efficient learning. In Chapters 3–5, we provide a detailed explanation, experimental results, and thorough discussions of the proposed methods. Finally, in Chapter 6, we conclude the dissertation and discuss the future directions of the research.

Chapter 2

Background

In this chapter, we will discuss a broad summary of object recognition and a variety of types of popular weak supervision. In addition, we provide explanations of the generic preliminary algorithms for a better understanding of the dissertation.

2.1 Object Recognition

Object recognition is one of the most important and interesting tasks in computer vision. The goal of object recognition is to identify objects in images or videos. The recognition of an object can be performed with various forms of output: the existence of each object class (classification), localization with bounding boxes (object detection), localization with pixel-level masks (semantic segmentation, instance segmentation), and so on. Object recognition has a broad practicality to real-world applications, such as autonomous driving [22], automated medical diagnosis [58, 59], and human-robot interaction [60, 61]. Image classification is the representative task for object recognition, which is a study of interest not only in computer vision but also in general deep learning and machine learning society. In this dissertation, we focus on the object localization tasks rather than image classification, so that we briefly review the overview of representative tasks in the object localization fields in the following chapters.

Object Detection Object detection aims at localizing instances of objects from the given object classes in the form of bounding boxes. One of the most popular pioneering deep learning approaches to object detection is Region-based Convolutional Neural Network (R-CNN) [16] based model. R-CNN first computes class-agnostic region proposals, which can be obtained by proposal generators such as selective search algorithm [62], and conduct classification and localization on the computed region proposals. R-CNN then extracts a feature vector for each proposal by warping and passing the proposals to several convolutional layers. The class of each proposal is predicted by passing the feature vector to the trained Support Vector Machines (SVMs), which provides class-specific scores for each proposal. The bounding box coordinates are predicted by the trained bounding box regression layers in parallel with the classification. However, the inference of R-CNN is super slow due to the complicated multi-stage processes. To address this, Fast R-CNN [63] and Faster-RCNN [15] are proposed.

The object proposals provide meaningful information for the location of the target object, but they are class-agnostic, and their box coordinates tend to be noisy, so that the proposal box is not accurately fit to the target object. On the obtained object proposals, R-CNN based models commonly pass the proposals to two heads: classification head and bounding box regression head. The classification head computes the class prediction p^* of the given proposal, which is trained with the cross-entropy loss. The bounding box regression head computes the displacement of each bounding box coordinate to fit the object, because the coordinates of proposals tend to be noisy. The regression head regresses the offsets $t = (t_x, t_y, t_w, t_h)$ for each coordinate of the box, and the final localization bounding box is obtained by shifting each coordinate of proposal box using the computed offset t.

R-CNN based models require two or more stages, including the object proposal computation step and processing two heads mentioned above, so the inference speed is not satisfactory. Therefore, one-stage methods for object detection are proposed.

YOLO [64] directly predicts bounding boxes from image pixels by reformulating the object detection task to the regression task. Lin *et al.* [65] propose focal loss to reduce the imbalance between foreground and background classes during the training of the single-stage object detectors. Focal loss is based on the classical cross-entropy loss, but it reduces the weights of the loss computed from well-predicted examples.

Recently, many methods based on Transformer [66] architectures have been proposed. DETR [17] is the pioneering work of Transformer for the object detection task. With this successful introduction of Transformer into object detection, numerous following works have emerged [67, 68, 69].

Semantic Segmentation Semantic segmentation aims at partitioning image regions into segments of each object class. Semantic segmentation can be formulated as a pixel-level classification problem with semantic object class labels. Even before the development of deep learning, classical approaches were actively studied, such as region growing [70], k-means clustering [71], and graph cuts [72]. Deep learning models have recently produced a new generation of semantic segmentation methods with remarkably improved performance, leading to a paradigm change in the industry. Semantic segmentation based on deep learning is popularly studied with various viewpoints: the construction of data, the choice of loss functions and network architectures, learning strategies, and so on.

One of the earliest deep learning efforts for semantic segmentation was proposed by Long *et al.* [73] utilizing a fully convolutional network (FCN). A FCN can take any size image and turn it into a segmentation map of the same size because it only has convolutional layers. In order to handle non-fixed sized input and output, all fully-connected layers are replaced by fully-convolutional layers. As a result, a spatial pixel-level classification segmentation map can be obtained.

With the success of FCN, convolutional encoder-decoder based architectures are popularly studied. An early research on semantic segmentation based on deconvolution layers (also known as transposed convolution layers) was published by Noh *et al.* [11].

The encoder-decoder based architecture is divided into two components: a convolutional encoder that encodes the semantic information for a given input image, and a deconvolutional network that outputs a map of pixel-wise class probabilities by using the encoded information by the encoder.

Another popular group of deep learning methods for semantic segmentation is pyramid-based network. The Feature Pyramid Network (FPN) introduced by Lin *et al.* [74] is one of the most popular models of this family. It was first proposed for the object detection task, but was later used for semantic segmentation as well. Deep convolutional neural networks inherently produce multi-scale features, therefore pyramidal hierarchy of deep features can be built with little additional expense. The FPN can consider rich information by merging shallow and deep features in a convolutional neural network.

DeepLab series [18, 19] are one of the most popular approaches in semantic segmentation. The success of DeepLab lies in the use of dilated convolution in the astrous spatial pyramid pooling (ASPP) module, and conditional random fields (CRFs) [75]. The multi-resolution information can be considered by using different rates of dilated convolution (*i.e.,* 1, 3, 6, 9) on deep features, resulting in an improved representation of context and semantics in an image. The produced segmentation maps by DeepLab models provide coarse segmentation of the target object, due to the low-resolution of deep features. This can be addressed by probabilistic graphical model, *i.e.,* CRFs [75]. Recently, attention-based approaches [76, 77] are popularly studied.

Instance Segmentation The goal of instance segmentation is to not only assign an object class to each pixel, but also separate individual objects. The representative trend on instance segmentation is to conduct object detection first, and to produce the segmentation in the predicted box. Mask R-CNN introduced by He *et al.* [20] is one of the most popular models of this family. Based on Faster R-CNN [15], Mask R-CNN brings an additional branch to predict the pixel-level mask in the box. Due to its efficiency and performance, many following works are proposed. Mask Scoring

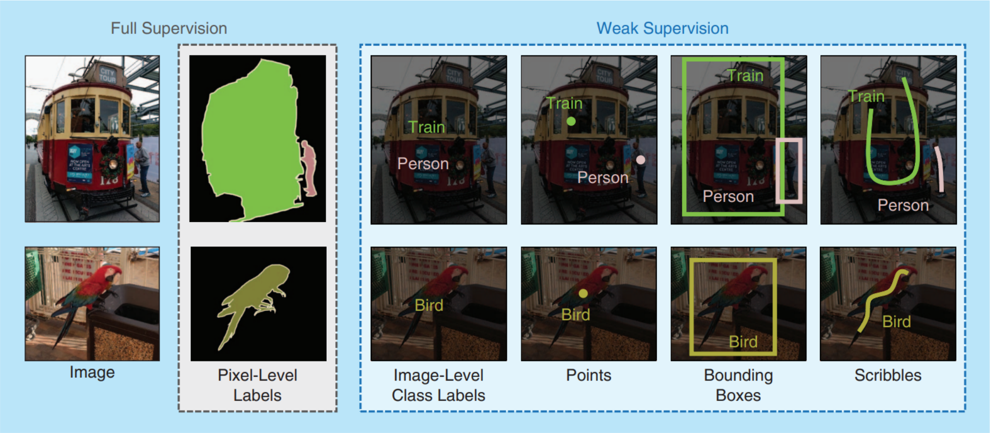


Figure 2.1: Examples of various form of weak supervision popularly used for weakly supervised semantic segmentation and instance segmentation. This figure is borrowed from Hong *et al.* [34]

R-CNN [78] improves the confidence estimate of the output of Mask R-CNN. It adds an additional head which predict the mask quality in terms of intersection-over-union (IoU). This approach enhances the performance of the instance segmentation task by giving higher priority to predictions which have stronger predictions, which is important for the COCO AP evaluation process [79]. To avoid the multi-stage processing of R-CNN based models, one-stage instance segmentation models such as YOLACT [80] and SOLO [81] also have been proposed.

Another research line for instance segmentation is clustering-based approach [82, 83, 84]. These approaches first conduct categorical labelling for each pixel in a given image similar to semantic segmentation, and group pixels corresponding to a single object instance using clustering algorithms. Although these methods can benefit from well-studied semantic segmentation techniques, they show relatively low performance compared to detection-based methods.

Table 2.1: Comparison of average annotation cost for an image (sec/img) for various form of weak supervision and fully supervised pixel-level masks. The average annotation times were borrowed from the work of Bearman *et al.* [21] and the work of Bellver *et al.* [55].

	Class	Point	Bounding Box	Scribble	Full
Cost (sec/img)	20.0	22.2	38.1	34.9	239.7

2.2 Weak Supervision

As we investigate in Chapter 2.1, deep learning has produced tremendous progress on the object recognition tasks. However, the success of deep learning on object recognition tasks has come at the cost of annotating thousands (or much more) of training images with explicit localization cues. In particular, for semantic segmentation, pixel-level annotation of images containing an average of 2.8 objects takes about 4 minutes per image [21]; and a single large (2048×1024) image depicting a complicated scene requires more than 90 minutes for pixel-level annotation [22].

The need for such expensive annotations can be sidestepped by weakly supervised learning, where a deep neural network is trained on images with some form of abbreviated annotation that is cheaper to obtain than explicit localization cues. Weakly supervised semantic segmentation methods can use scribbles [23], points [21], bounding boxes [24, 25, 26], or class labels [27, 28, 29, 30, 85] as annotations. A scribble is an arbitrary form of a line inside the target object, which is obtained through a single user stroke. The scribbles sparsely depict the location and extent of the target object. A point indicates a single point inside the target object, which can be considered as an extreme case of a scribble label. A bounding box provides a rectangular area that covers the target object tightly. These three forms of weak supervision provide the information about the location of the target object, but an image-level class label provides only the existence of an object corresponding to each semantic category in an image.

Figure 2.1 presents examples of various weak annotations (image-level class label, point, bounding box, and scribble) that are popularly used for weakly supervised semantic and instance segmentation. Table 2.1 compares the annotation cost for various forms of weak supervision and fully supervised pixel-level masks. We can see that weakly annotated labels are much cheaper to obtain than fully supervised labels. For example, class labels are about 12 times more efficient to obtain compared to fully supervised labels. Even the most expensive form of weak supervision (*i.e.,* bounding box) is over six times more efficient to obtain compared to fully supervised labels. Therefore, by developing methods that can train a deep neural network for semantic segmentation and instance segmentation with weakly supervised labels that can be easily obtained, the required data construction cost can be significantly reduced, and it can be used in more applications.

Of the various form of weak supervision, image-level class labels are the cheapest and most popular option, largely because the images in many public datasets are already annotated with class labels [3, 32], and automated web searches can also provide images with class labels [33, 34, 35]. Therefore, many researchers have studied weakly supervised learning methods using image-level class labels. Most weakly supervised semantic segmentation methods depend on attribution maps obtained from a trained classifier, such as a Class Activation Map (CAM) [36] or a gradient-based class activation map (Grad-CAM) [37]. An attribution map identifies the important, or discriminative, regions of an image on which the classifier has concentrated. But these regions tend to be relatively small, and most attribution maps do not identify the whole region occupied by the target object. Therefore, many researchers have tried to extend attributed regions to cover more of the target object, by manipulating either the image [86, 87, 88] or the feature map [27, 89, 90, 1]. CAM-based methods have an issue of coarse representation of the target object, because CAMs are computed based on the intermediate features of deep neural network that are down-sampled from the input image resolution. Therefore, CAMs do not represent the exact boundary of the target object. To address this, refinement techniques of CAMs have also been proposed [29, 2, 91].

However, methods using only class labels are known to suffer from the confusion between foreground and background cues, *i.e.,* spurious correlation problem. This is a fundamental problem that cannot be solved solely with the class labels; additional information is needed to learn to fully distinguish the foreground and background cues [46, 47, 50]. Researchers have thus studied various sources of additional guidance to separate the foreground and background cues, each with different pros and cons and different labeling-cost footprints. Image saliency [92, 93] is one of the most widely used ones [27, 50, 52, 94, 95, 96], for it naturally provides the prominent foreground object in the image in a class-agnostic fashion. However, saliency is not very effective for non-salient foreground objects (*e.g.,* low-contrast objects or small objects), and is limitedly applicable only to natural images. Low-level visual features like superpixels [53, 54], edges [97], object proposals [26, 25, 95], and optical flows [34, 33] have also been considered. Although these types of additional information are cost-effective, they tend to generate inaccurate object boundaries because such low-level information does not consider semantics associated with the object class. Moreover, class labels provide no help in separating different objects of the same class, which is the goal of instance segmentation.

Therefore, the bounding box label is more suitable for learning instance segmentation than image-level class labels, because bounding boxes provide information about individual objects and their locations. Bounding box annotation takes about 38.1 seconds per image [55], which is still attractive than constructing pixel-level masks. Many researchers have tackled semantic segmentation [98, 24, 25, 99] and instance segmentation [24, 100, 101, 102, 103] using bounding box annotations as a search space in which a class-agnostic object mask can be found by an off-the-shelf object mask generator. These are mostly based on GrabCut [56] or multiscale combinatorial grouping (MCG) [57]. Those mask generators operate on the low-level information of images, such as the color or brightness of pixels, and this limits the quality of the resulting mask. Thus, applying these mask generators to bounding box annotations requires additional steps such as estimating what proportion of the pixels in a bounding box belong to the corresponding object [25, 99], iterative refinement of an estimated mask [98], and auxiliary attention modules [99].

2.3 Preliminary Algirothms

This section provides general preliminary algorithms for the understanding of the following chapters. The specific preliminary algorithms for each chapter are included in the corresponding chapter.

2.3.1 Attribution Methods for Image Classifier

As mentioned in the earlier chapter, most weakly supervised semantic segmentation methods depend on attribution maps obtained from a trained classifier, such as a Class Activation Map (CAM) [36] or a Grad-CAM [37]. A class activation map (CAM) [36] identifies regions of an image focused by a classifier. The CAM is based on a convolutional neural network with global average pooling (GAP) before its final classification layer. This is realized by considering the class-specific contribution of each channel of the last feature map to the classification score. Given a classifier parameterized by $\theta = \{\theta_f, w\}$ where $f(\cdot;\theta_f)$ is the feature extractor prior to GAP, and w is the weight of the final classification layer, a CAM of the class c is obtained from an image x as follows:

$$\texttt{CAM}(x;\theta) = \frac{\mathbf{w}_c^\intercal f(x;\theta_f)}{\max \mathbf{w}_c^\intercal f(x;\theta_f)}, \tag{2.1}$$

where max(·) is the maximum value over the spatial locations for normalization.

The CAM is simple, easy to implement, and has powerful localization ability. However, the CAM can be applicable only for a specific model: a convolutional neural network with global average pooling (GAP), followed by only a single fully-connected classification layer. As such, the CAM operates only on the classification task, and even in the classification task, it operates on very limited architecture.

Selvaraju *et al.* [37] develop the gradient-based class activation map (Grad-CAM), which can be considered as a generalization of CAM [36]. Grad-CAM discovers the class specific contribution of each hidden unit to the classification score from gradient flows. Since gradient flow is very general in most deep neural networks, Grad-CAM can be applied to any differentiable architectures and any differentiable tasks. Grad-CAM is computed as follows: we first compute the gradients of the target class score with respect to any intermediate feature, and then sum the feature maps along the channel axis, weighted by these gradients. We can express Grad-CAM for each target class c as follows:

$$\text{Grad-CAM}^{\text{c}} = \text{ReLU}(\sum_k f(x;\theta_f)_k \times \frac{\partial S^c}{\partial f(x;\theta_f)_k}), \tag{2.2}$$

where $f(x;\theta_f)_k \in \mathbb{R}^{w \times h}$ is the k^{th} channel of the feature map $f(x, \theta_f)$, and S^c is the classification score of class c.

As these gradient-based attribution methods gain popularity, advanced methods have been proposed such as Grad-CAM++ [104], Score-CAM [105], and Relevance-CAM [106]. Because of their localization ability and simplicity, the CAM-based attribution methods are widely used for weakly supervised semantic segmentation [27, 33, 107, 49, 108] and weakly supervised object localization [1, 109, 89]. However, CAMs are known to have two major drawbacks that can be shown in Figure 2.2. First, CAMs identify only small, discriminative regions of the target object because all the regions of the target object do not necessarily contribute to the classification. Second, CAMs are obtained from the intermediate features of the deep neural network, which are down-sampled from the original image resolution. Therefore, CAMs represent the target object coarsely and do not depict the boundary of the target object accurately. These problems make it difficult for CAMs to be used as segmentation itself. There-

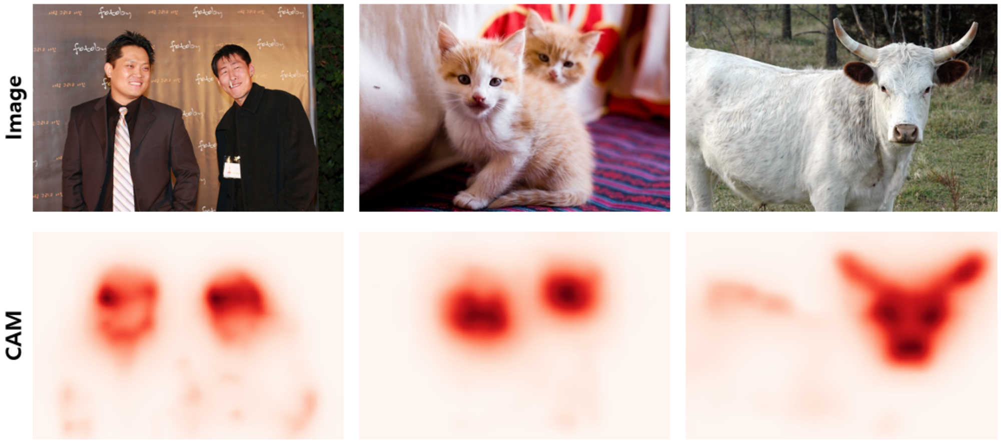


Figure 2.2: Examples of the class activation maps (CAMs) for 'person' class (*left*), 'cat' class (*middle*), and 'cow' class (*right*).

fore, many studies have been conducted to refine the CAMs toward more complete segmentation, which will be discussed in the next chapter.

2.3.2 Refinement Techniques of Localization Maps

In this chapter, we take a closer look at three representative refinement techniques to obtain accurate segmentation from coarse localization maps: Deep Seeded Region Growing (DSRG) [91], AffinityNet [29], and Inter-Pixel Relation Network (IRNet) [2]. These methods commonly obtain an initial seed from the CAM and learn the relationship between pixels from the initial seed. Using this learned relationship between pixels, they propagate the score of the confident regions of the initial seed, which have initially high CAM scores, to the neighboring ambiguous region, which have initially low CAM scores.

DSRG [91] first obtains the initial foreground cues from the discriminative object regions by applying a hard (high) threshold to the CAM. They also obtain background cues by utilizing the saliency detection method [110]. By combining foreground and background cues, they obtain initial seed cues. The initial seeds are sparse pseudo ground truth segmentation maps because they have ambiguous regions where neither the foreground scores nor the background scores are confident. The goal of DSRG is to gradually assign pseudo segmentation labels to those ambiguous regions during training the segmentation network. To train the segmentation network, the balanced seeding loss l_{seed} is used as follows:

$$l_{seed} = -\frac{1}{\sum\limits_{c\in\mathcal{C}} |S_c|} \sum_{c\in\mathcal{C}} \sum_{u\in S_c} \log H_{u,c} - \frac{1}{\sum\limits_{c\in\overline{\mathcal{C}}} |S_c|} \sum_{c\in\overline{\mathcal{C}}} \sum_{u\in S_c} \log H_{u,c}, \tag{2.3}$$

where $\mathcal{C}$ is the set of foreground classes that are present in the given image, which can be obtained from image-level class labels, and $\overline{\mathcal{C}}$ is the background class. S_c is the set of locations that are considered as class c in the pseudo ground truth, and $H_{u,c}$ is the predicted probability of class c at the location of u on the segmentation map H.

In the conventional approach, the pseudo ground truth S is fixed during the training of the segmentation network, but DSRG updates S in an online manner, based on the predicted segmentation map H. Starting from the initially confident regions, DSRG visits 8-connectivity neighborhoods of these confident pixels iteratively. If an ambiguous pixel u is connected to the confident regions with 8-connectivity neighborhoods, and the pixel u has a sufficiently high predicted segmentation score, then the pixel u is now included into the pseudo ground truth S. By iteratively repeating this process during each iteration of segmentation training, the pseudo ground truth S gradually covers more regions of the target object, resulting in a better pseudo segmentation label.

Together with this seeding technique, constrain-to-boundary loss is used to modeling precised object boundary by considering spatial and color information of the input image. This can be realized by conditional random fields (CRFs) [75]. During the training, we apply CRFs to the produced segmentation map H, resulting in $Q(H)$. The constrain loss $l_{constrain}$ optimizes the KL-divergence between the segmentation map H and the results of the CRFs, as follows:

$$l_{constrain} = \sum_{c\in\mathcal{C}} \sum_{u\in S_c} Q_{u,c}(H) \log \frac{Q_{u,c}(H)}{H_{u,c}}, \tag{2.4}$$

where $Q_{u,c}$ is the location u of the CRF map Q for the class c.

AffinityNet [29] also starts with the initial sparse pseudo ground truth S. The aim of Affinitynet is to learn class-agnostic semantic affinities between the pair of neighboring pixels on the training image. The semantic affinity is computed on the feature-level. For location indices i and j on the feature map F, the semantic affinity W_{ij} between the location i and the location j is defined as follows:

$$W_{ij} = \exp\{-||f_i - f_j||_1\}. \tag{2.5}$$

AffinityNet learns W_{ij} from the approximated binary affinity labels W^*_{ij}. For two pixels at the location i and j, the affinity label between i and j, W^*_{ij}, is assigned to 1 if they have the same pseudo segmentation labels, and assigned to 0 if they have different labels. The set of location pairs used for training is defined as $\mathcal{P}$. It contains a set of coordinate pairs within the Euclidean distance threshold γ. Then, the $\mathcal{P}$ is divided into the set of positive pairs $\mathcal{P}^+$ and the set of negative pairs $\mathcal{P}^-$ as follows:

$$\mathcal{P}^+ = \{(i,j)|(i,j) \in \mathcal{P}, W^*_{ij} = 1\}, \quad \mathcal{P}^- = \{(i,j)|(i,j) \in \mathcal{P}, W^*_{ij} = 0\},$$
$$\text{where } \mathcal{P} = \{(i,j)|d(x_i, x_j) < \gamma, i \neq j\}.$$

Here, x_i is the pixel coordinate of the location i.

Finally, AffinityNet is trained to make W_{ij} to produce 1 for $(i,j) \in \mathcal{P}^+$ and 0 for $(i,j) \in \mathcal{P}^-$, with the loss as follows:

$$\mathcal{L} = \mathcal{L}^+ + \mathcal{L}^-, \text{where}$$
$$\mathcal{L}^+ = -\frac{1}{|\mathcal{P}^+|} \sum_{(i,j)\in\mathcal{P}^+} \log W_{ij}, \quad \mathcal{L}^- = -\frac{1}{|\mathcal{P}^-|} \sum_{(i,j)\in\mathcal{P}^-} \log(1 - W_{ij})$$

AffinityNet refines CAMs of training images using the learned affinity W_{ij} by a random walk. The affinity W_{ij} provides the transition probability matrix used for the random walk. The confident scores of the initial CAMs are iteratively propagated to the semantically similar regions, resulting in the improved segmentation labels.

IRNet [2] has a similar concept to AffinityNet [29], but it has a different scheme to learn the semantic affinity. Because of its powerful performance and well-implemented code, IRNet [29] have been used as the off-the-shelf refinement technique in many recent methods, such as CONTA [48], AMN [111]. Compared to AffinityNet considers all pairs of pixels to learn the affinity, IRNet employs the multiple instance learning technique to consider the relationship between pixels. The semantic affinity used in IRNet is expressed as the existence of the boundary of class. More specifically, IRNet computes boundary maps $\mathcal{B} \in [0,1]^{w \times h}$, which represents the boundaries between two different classes. Based on the produced $\mathcal{B}$, the semantic affinity W_{ij} is computed as follows:

$$W_{ij} = 1 - \max_{k \in \Pi_{ij}} \mathcal{B}(x_k), \tag{2.6}$$

where x_k is the pixel located at k, Π_{ij} is the set of pixels presenting on the line connecting two pixels x_i and x_j. The semantic affinity is learned through the loss presented in Eq. 2.3.2, with the positive and negative pairs denoted in Eq. 2.3.2. Similar to AffinityNet [29], the learned semantic affinity is used for random walk to refine the CAMs.

Chapter 3

Learning with Image-Level Class Labels

3.1 Introduction

Understanding the semantics of an image and recognizing objects in it are vital processes in computer vision systems. Although deep neural networks (DNNs) have facilitated tremendous progress in both tasks [18, 76, 19, 15, 64, 65], it has come at the cost of annotating thousands of training images with explicit localization cues. The need for such expensive annotations is sidestepped by weakly supervised learning, in which a DNN is trained on images with some form of abbreviated annotation that is cheaper than explicit localization cues. Weakly supervised semantic segmentation methods can use scribbles [23], points [21], bounding boxes [24, 25, 26], or class labels [27, 28, 29, 30, 85] as annotations. The last of these are the cheapest and most popular option, largely because the images in many public datasets are already annotated with class labels [3, 32], and automated web searches can also provide images with class labels [33, 34, 35].

Most weakly supervised semantic segmentation and object localization methods depend on attribution maps obtained from a trained classifier, such as a Class Activation Map (CAM) [36] or a Grad-CAM [37]. An attribution map identifies the important, or discriminative, regions of an image on which the classifier has concentrated. But these regions tend to be relatively small, and most attribution maps do not identify the whole region occupied by the target object. Therefore, many researchers have tried to extend attributed regions to cover more of the target object, by manipulating either the image [86, 87, 88] or the feature map [27, 89, 90, 1].

In this chapter, we discuss three studies on weakly supervised semantic segmentation using image-level class labels. In Chapter 3.3, we introduce AdvCAM [112, 113], a new manipulation method for extending the attributed regions of a target object. In Chapter 3.4, we theoretically discuss the reason why the CAMs identify only small regions of the target object in the view of information bottleneck, and introduce RIB [49] to address the problem. Lastly, we will introduce FickleNet [27], a stochastic inference technique, but we will discuss this in Chapter 3.2 briefly for the conciseness.

3.2 Related Work

In this chapter, we first introduce our method FickleNet, the stochastic inference approach for weakly supervised semantic segmentation using class labels. We then discuss other recent approaches related to the methods proposed in this dissertation.

3.2.1 FickleNet: Stochastic Inference Approach

To address the problem of CAMs identifying only discriminative regions of the target object, we propose FickleNet, which is a stochastic inference technique generating a variety of localization maps from a single image using random combinations of hidden units in a convolutional neural network, as shown in Figure 3.1(a). Starting with a feature map created by a generic classification network such as VGG-16 [114], FickleNet chooses hidden units at random for each sliding window position, which corresponds to each stride in the convolution operation, as shown in Figure 3.1(b). This process is simply realized by the dropout method [115]. Selecting all the available hidden units in a sliding window position (the deterministic approach) tends to produce

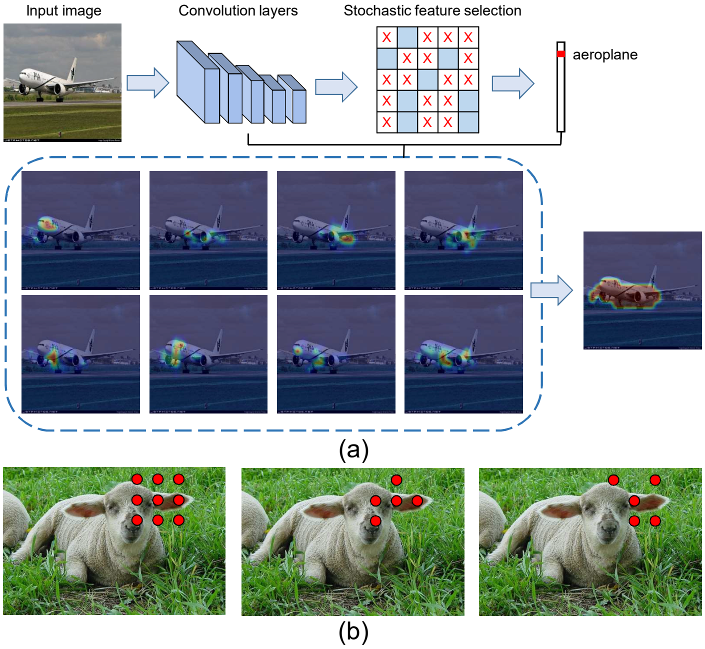


Figure 3.1: (a) FickleNet allows a single network to generate multiple localization maps from a single image. (b) Conceptual description of hidden unit selection. Compared to selecting all hidden units (deterministic, *left*), randomly selected hidden units (stochastic, *center* and *right*) can provide more flexible combinations.

a smoothing effect that confuses foreground and background, which can result in both areas being activated or deactivated together. However, random selection of hidden units (the stochastic approach) produces regions of different shapes which can delineate objects more sharply. Since the patterns of hidden units randomly selected by FickleNet include the shapes of the kernel of the dilated convolution with different dilation rates, FickleNet can be regarded as a generalization of dilated convolution, but FickleNet can potentially match objects of different scales and shapes using only a single network because it is not limited to a square array of hidden units, whereas dilated convolution requires networks with different dilation rates just to scale its kernel.

The selection of random hidden units at each sliding window position is not an operation that is optimized at the CUDA level in common deep-learning frameworks

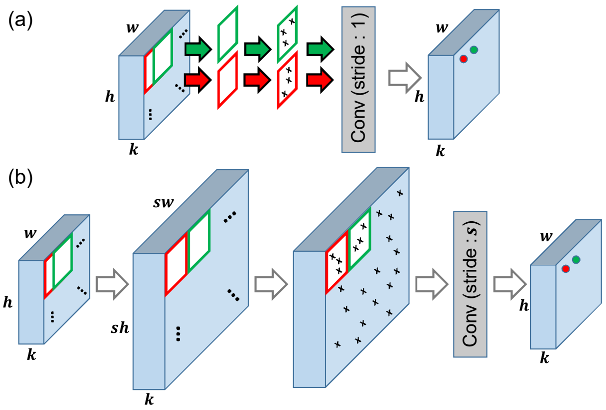


Figure 3.2: (a) Naive implementation of FickleNet, which requires a dropout and convolution function call at each sliding window position (the red and green boxes). (b) Implementation using map expansion: convolution is now performed once with a stride of s. The input feature map is expanded so that successive sliding kernels (the red and green boxes) do not overlap.

such as PyTorch [116]. Thus, a naive implementation of FickleNet, in which random hidden units are selected at each sliding window position and then convolved, would require a large number of iterative operations. However, we can use the optimized convolution functions provided by deep-learning frameworks, if we expand the feature maps before making the random selection of hidden units. The maps need to be expanded sufficiently to prevent successive sliding window positions from overlapping. We can then apply dropout in the spatial axis of the expanded feature maps, and perform a convolution operation with a stride equal to the kernel size. This saves a significant amount of time without much increase in GPU memory usage, because the number of parameters to be back-propagated remains constant. The illustration of this expansion technique is shown in Figure 3.2.

While many existing networks use stochastic regularization in their training process (e.g. Dropout [115]), stochastic effects are usually excluded from the inference process. However, our inference process contains random processes and thus produces a variety of localization maps. The pixels that were allocated to a specific class with high scores in each localization map are discovered, and those pixels are aggregated into a single localization map. The localization map obtained from FickleNet is utilized as pseudo-labels for the training of a segmentation network.

3.2.2 Other Recent Approaches

Image-level Processing: Image-level hiding and erasure have been proposed [88, 86, 87] as ways of preventing a classifier from focusing exclusively on the discriminative parts of objects. Hide-and-Seek [88] hides random regions of a training image, forcing the classification network to seek other parts of the object. However, the process of hiding random regions does not consider the semantics and sizes of objects. Adversarial Erasing [86] starts with a single small region in the object, and then drives the classification network to discover a sequence of new and complement any object regions by erasing the regions that have already been found. Although it can progressively expand regions belonging to an object, it requires multiple classification networks to perform the repetitive classification and erasure steps. The Guided Attention Inference Network (GAIN) [87] has a CAM which is trained to erase regions in a way that deliberately confuses the classifier. This CAM has to be large enough to cover an entire object. However, the classifier mainly reacts to high activation, and so it can become confused if an object's only discriminative parts are erased.

Feature-level Processing Feature-level processing can be used to expand the regions activated by a CAM. Adversarial complementary learning [89] and two-phase learning [117] use a classifier to identify the discriminative parts of an object and erase them based on features. A second classifier then is trained to find the complementary parts of the object from those erased features. This is an efficient technique which operates at a relatively high level. However, it has a similar drawback to image-level erasure, in that a second classifier and training step are essential for those methods, which may cause a suboptimal performance. In addition, features whose discriminative parts are erased can confuse the second classifier, which may not be correctly trained. Pyramid Grad-CAM [59] considers multi-layer features for multi-scale context. Wei *et al.* [108] and Lee *et al.* [59] consider the target object in several contexts by combining multiple attribution maps from differently dilated convolutions or from different layers of a DNN.

Improved Learning Technique: The improved training technique for the deep neural network also have been actively studied. Wang *et al.* [107] use equivariance regularization during the training of their classifier so that the attribution maps obtained from differently transformed images are equivariant to those transformations. Chang *et al.* [30] improve feature learning by using latent semantic classes that are sub-categories of annotated parent classes, which can be pseudo-labeled by clustering image features. Fan *et al.* [118] and Sun *et al.* [94] capture information shared between several images by considering cross-image semantic similarities and differences. Zhang *et al.* [48] analyze the co-occurrence context problem in multi-label classification and propose context adjustment (CONTA) to remove the confounding bias, resulting in a CAM seed free of spurious correlations.

Region Growing: Region growing can be used to expand the localization map produced by a CAM, which initially identifies just the small discriminative part of an object. AffinityNet [29] learns pixel-level semantic affinities, which identify pixels belonging to the same object, under the supervision of an initial CAM, and then expands the initial CAM by a random walk with the transition matrix computed from semantic affinities. However, the learning of semantic affinities requires an additional network, and the outcome depends heavily on the quality of the CAM. Seed, Expand, and Constrain (SEC) [119] uses a new type of loss function to expand the localization map and constrain it to object boundaries using a conditional random field (CRF) [75]. Deep seeded region growing (DSRG) [91] refines initial localization maps during the training of its segmentation network, so that DSRG does not require additional networks to grow regions. IRN [2] extend the object region to semantically similar areas by a random walk. BEM [120] synthesizes a pseudo boundary from a CAM and then uses a similar propagation with IRN [2].

3.3 Anti-Adversarially Manipulated Attribution

3.3.1 Adversarial Attack

An adversarial attack attempts to fool a DNN by presenting it with images that have been manipulated with intent to deceive. Adversarial attack can be applied to classifiers [38, 121], semantic segmentation networks [122], or object detectors [123]. Not only the predictions of a DNN, but also the attribution maps can be altered by adversarial image manipulation [124] or model parameter manipulation [125]. These types of attacks try to make the DNN produce a spurious attribution map that identifies a wrong location in the image, or a map that might have been obtained from a completely different image, without significantly changing the output of the DNN.

An adversarial attack on a classifier aims to find a small pixel-level perturbation that can change its decision. In other words, given an input x to the classifier, the adversarial attack aims to find the perturbation n that satisfies $\texttt{NN}(x) \neq \texttt{NN}(x+n)$, where $\texttt{NN}(\cdot)$ is the classification output from the DNN. A representative method [38] of constructing n for an attack starts by constructing the vector normal to the decision boundary of $\texttt{NN}(x)$, which can be realized by finding the gradients of $\texttt{NN}(x)$ with respect to x. A manipulated image x' can then be obtained as follows:

$$x' = x - \xi \nabla_x \texttt{NN}(x), \tag{3.1}$$

where ξ determines the extent of the change to the image. This process can be understood as performing gradient descent on the image. PGD [39], which is a popular method of adversarial attack, performs the manipulation of Eq. 3.1 iteratively.

3.3.2 Proposed Method

Adversarial Climbing: AdvCAM is an attribution map obtained through adversarial climbing, which is an anti-adversarial technique that manipulates the image so as to increase the classification score of that image, with the result that the classifier identifies more regions of objects. This is the reverse of an adversarial attack based on Eq. 3.1, which manipulates the image to reduce the classification score. Inspired by PGD [39], iterative adversarial climbing of the initial image x^0 can be performed using the following relation:

$$x^t = x^{t-1} + \xi \nabla_{x^{t-1}} y_c^{t-1}, \tag{3.2}$$

where t $(1 \leq t \leq T)$ is the adversarial step index, x^t is the manipulated image at the $t-$th step, and y_c^{t-1} is the classification logit of x^{t-1} for class c.

This process makes the previously non-discriminative yet relevant features become more involved in the classification. Thus, the CAMs obtained from successive images manipulated by the iteration can be expected to identify an increasing amount of the region of the target object. We produce a localization map $\mathcal{A}$ which encapsulates the results of the iteration by aggregating the CAMs obtained from the manipulated images at each iteration t, as follows:

$$\mathcal{A} = \frac{\sum_{t=0}^{T} \mathtt{CAM}(x^t)}{\max \sum_{t=0}^{T} \mathtt{CAM}(x^t)}. \tag{3.3}$$

How can Adversarial Climbing Improve CAMs? The connection between a classification logit y_c and a CAM, *i.e.* $y_c =$ GAP($\mathtt{CAM}$) [89], infers that adversarial climbing increases y_c, and thus the CAM. In this process, features involved in classification are enhanced. To provide a better understanding how adversarial climbing generates a denser CAM, we consider two questions: ① Can non-discriminative features be enhanced? ② Are those enhanced features class-relevant from a human point of view?

① Can non-discriminative features be enhanced?: One might think that changing a pixel with a large gradient primarily enhances discriminative features. This pixel

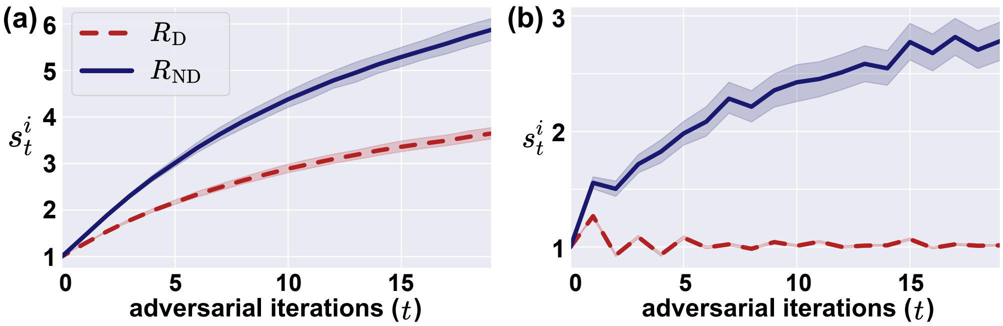


Figure 3.3: Distributions of the pixel amplification ratio s_t^i for $i \in R_{\mathrm{D}}$ and $i \in R_{\mathrm{ND}}$ for 100 images, (a) without regularization and (b) with regularization.

change affects many features due to the receptive field. However, not all the affected features are necessarily discriminative. We support this analysis empirically. We define the discriminative region $R_{\mathrm{D}} = \{i | \mathtt{CAM}(x^0)_i \geq 0.5\}$ and the non-discriminative region $R_{\mathrm{ND}} = \{i | 0.1 < \mathtt{CAM}(x^0)_i < 0.5\}$, where i is the location index. The pixel amplification ratio s_t^i is $\mathtt{CAM}(x^t)_i / \mathtt{CAM}(x^0)_i$ at location i and step t. Figure 3.3(a) shows that adversarial climbing makes both $s_t^{i \in R_{\mathrm{D}}}$ and $s_t^{i \in R_{\mathrm{ND}}}$ grow, but enhances non-discriminative features more than discriminative ones, resulting in a denser CAM.

② Are those enhanced features class-relevant from a human point of view? We now consider whether the highlighted non-discriminative features are class-relevant from a human point of view. Moosavi *et al.* [126] argued that a loss landscape that is sharply curved with respect to input makes a NN vulnerable to adversarial attack. Researchers have subsequently shown that a flattened loss landscape, obtained by reducing the curvature of the loss surface [126] or encouraging the loss to behave linearly [127], can improve the robustness of a NN. Systems which are robust in this sense have been shown to produce features that align better with human perception and operate in a easier way to understand [128, 129, 130].

By the same token, we can expect that images manipulated by adversarial climbing will produce features that align with human perception well because the curvature of loss surface affected by adversarial climbing is small. To support this, we visualize the

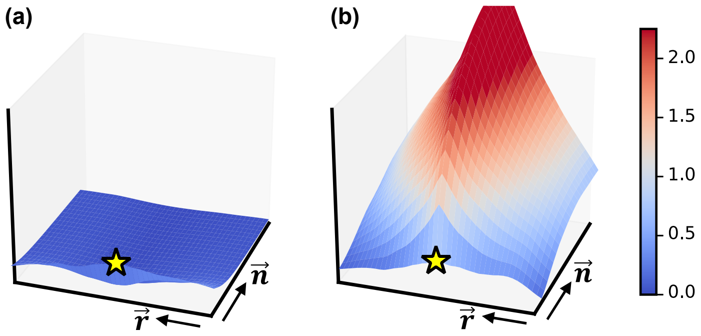


Figure 3.4: Loss landscapes by manipulating images with weighted sums of the normal vector $\vec{n}$ and a random vector $\vec{r}$ for (a) adversarial climbing and (b) adversarial attack. The yellow star corresponds to the original image.

loss landscape of our trained classifier, following Moosavi *et al.* [126]: we obtain a manipulation vector $\vec{n}$ and a random vector $\vec{r}$ from the classification loss ℓ computed from an image. We determine the surfaces of classification loss values computed from images, manipulated by a vector which is interpolated between $\vec{n}$ and $\vec{r}$ using a range of interpolation ratios. The loss landscape obtained by adversarial climbing (Figure 3.4(a)) is much more flatten than that obtained by adversarial attacking (Figure 3.4(b)). Therefore, we can legitimately expect it to increase the attribution of features relevant to the class from a human point of view, resulting in a better CAM.

Regularization: Even if the loss surface obtained by adversarial climbing is reasonably flat, too much repetitive adversarial manipulation may cause regions corresponding to objects in the wrong class to be activated, or increase the attribution scores of the regions that already have high scores. We address this by (i) suppressing the logit values associated with other classes and (ii) restricting high attributions on discriminative regions of the target object.

Suppressing Other Classes: In an image, objects of different classes can mutually increase logit values. For example, since a chair and a dining table mainly occur together in an image, a NN may infer an increased logit value for the chair from the region of the table. We thus add regularization that reduces logit values for all classes except c.

Restricting High Attributions: As mentioned earlier, adversarial climbing increases the attribution scores for both discriminative and non-discriminative regions in the feature map. However, the growth of attribution scores for discriminative regions is problematic for two reasons: 1) it prevents new regions from being additionally attributed to the classification score, and 2) if the maximum value of the attribution score increases during adversarial climbing, the normalized scores of the remaining area may decrease. Please see the blue boxes in Figure 3.5(b).

Therefore we limit the attribution scores in regions that already have high scores during adversarial climbing, so the attribution scores of those regions remain similar to that of x^0. We realize this scheme by introducing a restricting mask $\mathcal{M}$ that contains the regions whose attribution scores of $\texttt{CAM}(x^{t-1})$ are higher than the threshold τ. More specifically, $\mathcal{M}$ can be represented as follows:

$$\mathcal{M} = \mathbb{1}(\texttt{CAM}(x^{t-1}) > \tau), \tag{3.4}$$

where $\mathbb{1}(\cdot)$ is an indicator function. An example mask $\mathcal{M}$ is shown in Figure 3.5(a).

We add the regularization term so that the values of the CAM corresponding to the regions of $\mathcal{M}$ are forced to equal to that of $\texttt{CAM}(x^0)$. With this regularization, $s_t^{i\in R_{\text{D}}}$ remains fairly constant but $s_t^{i\in R_{\text{ND}}}$ still grows during adversarial climbing (Figure 3.3(b)). Figure 3.3 shows that, adversarial climbing enhances non-discriminative features more than discriminative features (¡ 2×), and regularization makes this difference even larger (¿ 2.5×). Thus, new regions of the target object are found more effectively, resulting in a denser CAM (Figure 3.5(b)).

To apply regularization, we modify Eq. 3.2 as follows:

$$x^t = x^{t-1} + \xi \nabla_{x^{t-1}} \mathcal{L}, \quad \text{where} \tag{3.5}$$

$$\mathcal{L} = y_c^{t-1} - \sum_{k\in\mathcal{C}\setminus c} y_k^{t-1} - \lambda \left\| \mathcal{M} \odot \left|\texttt{CAM}(x^{t-1}) - \texttt{CAM}(x^0)\right| \right\|_1 . \tag{3.6}$$

$\mathcal{C}$ is the set of all classes, λ is a hyper-parameter that controls the influence of masking regularization, and $\odot$ is element-wise multiplication.

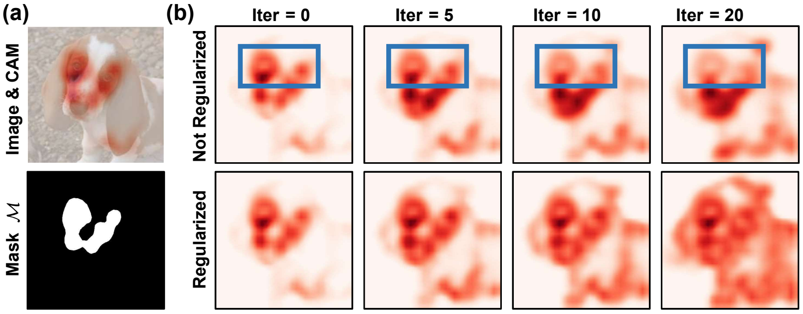


Figure 3.5: (a) An example image with its CAM and restricting mask $\mathcal{M}$. (b) The initial CAM, and CAMs after 5, 10 and 20 steps of adversarial climbing, with and without regularization.

Training Segmentation Networks Since CAM is obtained from down-sampled intermediate features produced by the classifier, it localizes the target object coarsely and cannot represent its exact boundary. Many methods of generating an initial seed for weakly supervised semantic segmentation construct a pseudo ground-truth by modifying their initial seeds using existing seed refinement methods [91, 29, 2]. For example, SEAM [107] and Chang *et al.* [30] use PSA [29]; and MBMNet [131] and CONTA [48] use IRN [2]. We also apply the seed refinement method to the coarse map $\mathcal{A}$. For weakly supervised learning, we use the resulting profiles as pseudo ground-truth for training DeepLab-v2, pre-trained on the ImageNet dataset [3]. For semi-supervised learning, we employ CCT [132], which uses IRN [2] to generate pseudo-ground truth masks; we replace these with our masks, constructed as just described.

3.3.3 Experiments

Dataset: We conducted experiments on the PASCAL VOC 2012 [32] dataset. The images in this dataset come with masks for fully supervised semantic segmentation, but we only used them for evaluation. In a weakly supervised setting, we trained our network on 10,582 training images provided by Hariharan *et al.* [133], which have image-level annotations. In a semi-supervised setting, we used 1,464 training images with pixel-level annotations and 9,118 training images with class labels, following previous works [27, 132, 108, 134]. We evaluated our results by calculating mean intersection-over-union (mIoU) values for 1,449 validation images and 1,456 test images. Since the labels for test images are not publicly available, the results for those images were obtained from the official PASCAL VOC evaluation server.

Reproducibility: We performed iterative adversarial climbing with $T = 27$ and $\xi = 0.008$. We set λ to 7 and τ to 0.5. To generate the initial seed, we followed the procedure of Ahn *et al.* [2], including the use of ResNet-50 [135]. For final segmentation, we used DeepLab-v2-ResNet101 [18] as the backbone network. We followed the default settings of [18] for training, which included cropping the images to 321×321 pixels. In a semi-supervised setting we used the same settings as Ouali *et al.* [132].

Quality of the Mask: Table 4.2 compares the initial seed and pseudo ground truth masks obtained from our method and from other recently published techniques. Both seeds and masks were generated from training images of the PASCAL VOC dataset. For initial seeds, we report the best results by thresholding with a range of threshold values to discriminate the foreground and background in the produced localization map $\mathcal{A}$, as following SEAM [107]. Our initial seeds are 6.8% better than the original CAMs [36], which provide a baseline, and this also outperforms the other methods. Note that Chang *et al.* [30] and SEAM [107] use Wide ResNet-38 [136], which provides better representation than ResNet-50 [135]. SEAM [107] also uses an auxiliary self-attention module that performs pixel-level refinement of the initial CAM by considering the relationship between pixels. We apply CRF, a widely used post-processing method, to the initial seeds of Chang *et al.* [30], SEAM [107], IRN [2], and our method. With the exception of SEAM, CRF improves the seed by more than 5% on average, but it improves the seed of SEAM only by 1.4%. We believe this is because the seed of SEAM is already refined by the self-attention module. Our seed after applying CRF is 5.3% better than that of SEAM.

We also compared pseudo ground truth masks, extracted after seed refinement, with

Table 3.1: Comparison of the quality of the initial seed and pseudo ground-truth with state-of-the-art methods in terms of mIoU on PASCAL VOC 2012 *train* images.

Method	Seed	+ CRF	Mask
Seed Refine with PSA [29]:			
PSA $_{\text{CVPR '18}}$ [29]	48.0	-	61.0
Chang *et al.* $_{\text{CVPR '20}}$ [30]	50.9	55.3	63.4
SEAM $_{\text{CVPR '20}}$ [107]	55.4	56.8	63.6
AdvCAM (Ours)	**55.6**	**62.1**	**68.0**
Seed Refine with IRN [2]:			
IRN $_{\text{CVPR '19}}$ [2]	48.8	54.3	66.3
MBMNet $_{\text{ACMMM '20}}$ [131]	50.2	-	66.8
CONTA $_{\text{NeurIPS '20}}$ [48]	48.8	-	67.9
AdvCAM (Ours)	**55.6**	**62.1**	**69.9**

existing methods. Most seed generator methods refine their generated localization maps with IRN [2] or PSA [29]. For a fair comparison, we produced pseudo ground truth masks using both these seed refinement techniques. Table 4.2 shows that our method outperforms the others by a large margin, whichever seed refinement technique is used.

Weakly Supervised Semantic Segmentation: Table 5.3 compares our method with other recently introduced weakly supervised semantic segmentation methods with various levels of supervision: fully supervised pixel-level masks ($\mathcal{P}$), bounding boxes ($\mathcal{B}$) or image class labels ($\mathcal{I}$), with and without salient object masks ($\mathcal{S}$). All the results in Table 5.3 were obtained using a ResNet-based backbone [135]. With image-level annotation alone, our method achieves mIoU values of 68.1 and 68.0 for the PASCAL VOC 2012 validation and test images respectively. This is significantly better than the other methods under the same level of supervision. In particular, the mIoU value for validation images is 4.6% higher than that for IRN [2], which is our baseline. CONTA [48], the best-performing method among our competitors, achieves an mIoU value of 66.1; but their method depends upon SEAM [107], which is known to outperform IRN [2]. If CONTA is implemented with IRN, the resulting mIoU value is 65.3, which is 2.8% worse than our method. Figure 4.4 presents examples of semantic masks produced by FickleNet [27], IRN [2], and our method.

Our method also outperforms other methods using auxiliary salient object mask supervision [92, 93] that provides exact boundary information of salient objects in an image, or extra web images or videos [94, 33]. The performance of our method is also comparable with that of methods [25, 24] that use bounding box supervision.

Semi-Supervised Semantic Segmentation: Table 3.3 compares the mIoU scores of our method on the PASCAL VOC validation and test images with those of other recent semi-supervised segmentation methods, which use 1.5K images with fully supervised masks and 9.1K images with weak annotations. All the methods in Table 3.3 were implemented on the ResNet-based backbone [135], except that daggered (†) methods which used the VGG-based backbone [114]. We achieve mIoU values of 77.8 and 76.9 for the PASCAL VOC 2012 validation and test images respectively, which is better than the other methods under the same level of supervision. Specifically, the performance of our method on the validation images was 4.6% better than that of CCT [132], which is our baseline. Our method even outperforms Song *et al.* [25] which uses bounding box labels for 9.1K images, instead of class labels. Figure 4.4 presents examples of semantic masks produced by CCT [132] and our method.

3.3.4 Discussion

Iterative Adversarial Climbing: We analyzed the effectiveness of the iterative adversarial climbing and regularization technique by evaluating the initial seed in terms of mIoU. Figure 3.7(a) shows the mIoU of the initial seed for each adversarial iteration. Initially, the mIoU rises steeply, with or without regularization; but without regularization the curves peaks around iteration 8.

To analyze this, we evaluate the truthfulness of the newly localized region at each

Table 3.2: Comparison of weakly supervised semantic segmentation performance on PASCAL VOC 2012 validation and test images.

Method	Sup.	*val*	*test*
Supervision: Image-level tags			
Li *et al.* ICCV '19 [137]	$\mathcal{I}, \mathcal{S}$	62.1	63.0
FickleNet CVPR '19 [27]	$\mathcal{I}, \mathcal{S}$	64.9	65.3
Lee *et al.* ICCV '19 [33]	$\mathcal{I}, \mathcal{S}, \mathcal{W}$	66.5	67.4
CIAN AAAI '20 [118]	$\mathcal{I}, \mathcal{S}$	64.3	65.3
Zhang *et al.* ECCV '20 [138]	$\mathcal{I}, \mathcal{S}$	66.6	66.7
Sun *et al.* ECCV '20 [94]	$\mathcal{I}, \mathcal{S}, \mathcal{W}$	67.7	67.5
IRN CVPR '19 [2]	$\mathcal{I}$	63.5	64.8
SSDD ICCV '19 [28]	$\mathcal{I}$	64.9	65.5
SEAM CVPR '20 [107]	$\mathcal{I}$	64.5	65.7
Chen *et al.* ECCV '20 [120]	$\mathcal{I}$	65.7	66.6
Chang *et al.* CVPR '20 [30]	$\mathcal{I}$	66.1	65.9
CONTA NeurIPS '20 [48]	$\mathcal{I}$	66.1	66.7
AdvCAM (Ours)	$\mathcal{I}$	**68.1**	**68.0**

$\mathcal{P}$—pixel-level mask, $\mathcal{I}$—image class, $\mathcal{B}$—box, $\mathcal{S}$—saliency, $\mathcal{W}$—web

adversarial climbing iteration in terms of the proportion of noise, which we define to be the proportion of pixels that are classified as foreground but are actually background. Without regularization, the proportion of noise rises steeply after some iterations as shown in Figure 3.7(b), which means that new regions tend to be in the regions of background. Regularization allows new regions of the target object to be found in as many as 30 adversarial steps, keeping the proportion of noise much lower than that of initial CAM. Figure 3.8 shows examples of attribution maps at each adversarial iteration with and without regularization.

Regularization Coefficient λ: It controls the influence of the masking technique that limits the attribution scores of the regions that already have high scores during

Table 3.3: Comparison of semi-supervised semantic segmentation methods on the PASCAL VOC 2012 *val* and *test* images.

Method	Training set	*val*	*test*
WSSL† [139]	1.5K $\mathcal{P}$ + 9.1K $\mathcal{I}$	64.6	66.2
MDC† [108]	1.5K $\mathcal{P}$ + 9.1K $\mathcal{I}$	65.7	67.6
Souly *et al.*† [140]	1.5K $\mathcal{P}$ + 9.1K $\mathcal{I}$	65.8	-
FickleNet† [27]	1.5K $\mathcal{P}$ + 9.1K $\mathcal{I}$	65.8	-
Song *et al.* [25]	1.5K $\mathcal{P}$ + 9.1K $\mathcal{B}$	71.6	-
Luo *et al.* [134]	1.5K $\mathcal{P}$ + 9.1K $\mathcal{I}$	76.6	-
CCT [132] (baseline)	1.5K $\mathcal{P}$ + 9.1K $\mathcal{I}$	73.2	-
AdvCAM (Ours)	1.5K $\mathcal{P}$ + 9.1K $\mathcal{I}$	**77.8**	**76.9**

$\mathcal{P}$−pixel-level mask, $\mathcal{I}$−image class label, $\mathcal{B}$−box, †− VGG backbone

Table 3.4: Effects of AdvCAM on different methods of generating the initial seed: mIoU of the initial seed (Seed) and of the pseudo ground truth mask (Mask), for the PASCAL VOC 2012 training images.

Method	Seed	Mask
Chang *et al.* [30]	50.9	63.4
+ AdvCAM	53.7 $_{+2.8}$	67.5 $_{+4.1}$
SEAM [107]	55.4	63.6
+ AdvCAM	58.6 $_{+3.2}$	67.2 $_{+3.6}$
IRN [29]	48.8	66.3
+ AdvCAM	55.6 $_{+6.8}$	69.9 $_{+3.6}$

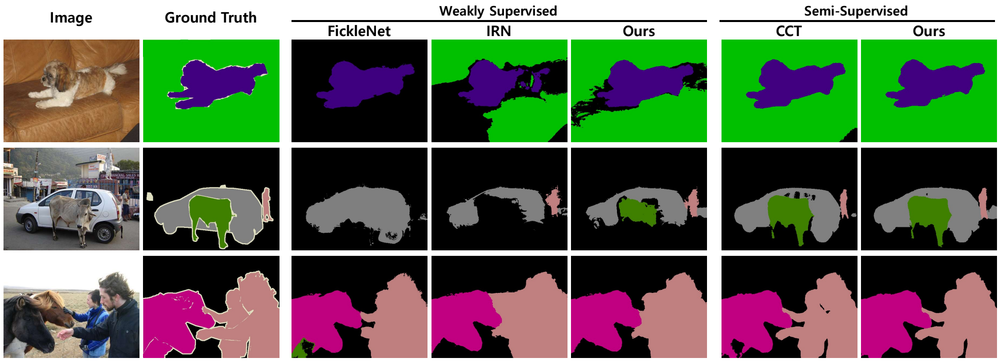


Figure 3.6: Examples of predicted semantic masks for PASCAL VOC *val* images in weakly and semi-supervised manner.

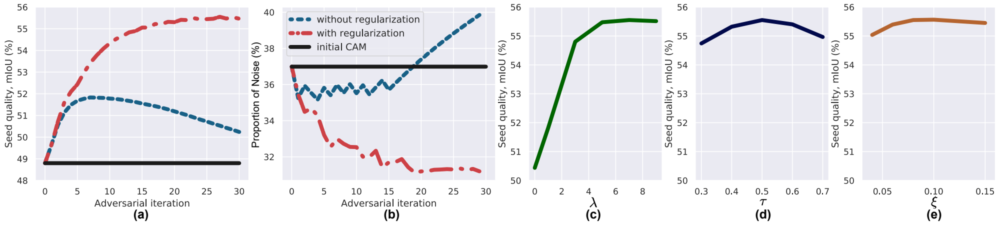


Figure 3.7: Effect of adversarial climbing and regularization on (a) the seed quality and (b) the proportion of noise. (c) Effect of the regularization coefficient λ. (d) Effect of the masking threshold τ. (d) Effect of the step size ξ.

adversarial climbing, in Eq. 3.6. Figure 3.7(c) shows the mIoU of the initial seed for different values of λ. When $\lambda = 0$, there is no regularization. Masking technique improves performance by more than 5% (50.43 for $\lambda = 0$ *vs.* 55.55 for $\lambda = 7$). The flattening of the curve after $\lambda = 5$ suggests that it is not difficult to select a good value of λ.

Masking Threshold τ: It controls the size of the restricting mask $\mathcal{M}$ in Eq. 3.4, determining how many pixels' attribution values will remain similar to that of the original CAM during adversarial climbing. Figure 3.7(d) shows the mIoU of the initial seed for different values of τ. This parameter is even less sensitive than λ: varying τ between 0.3 and 0.7 produces less than 1% change in mIoU.

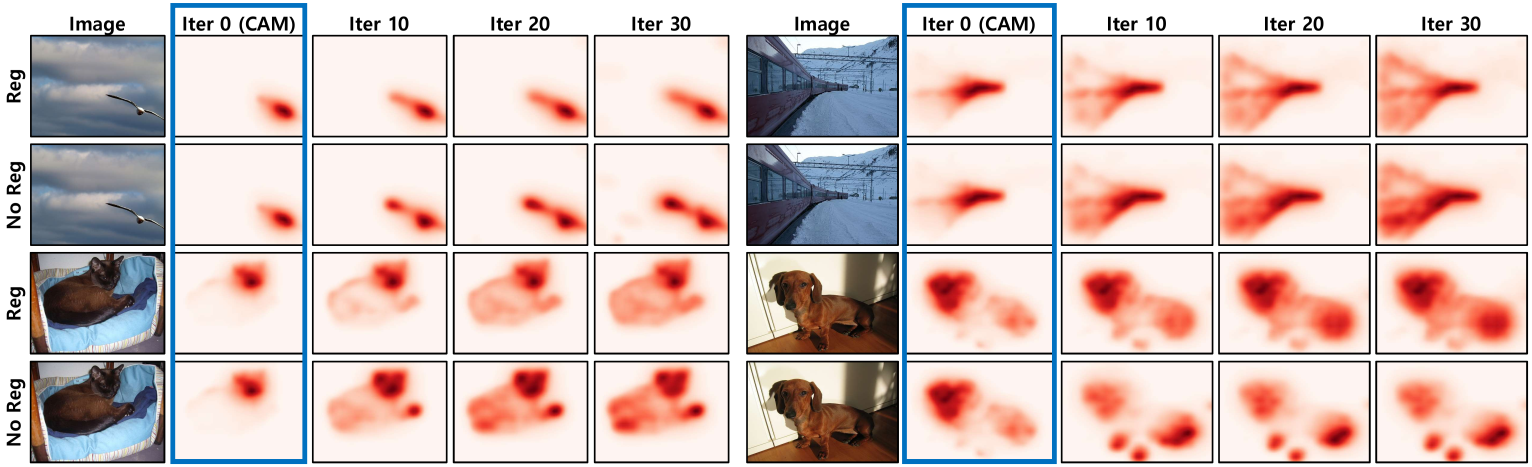


Figure 3.8: Examples of initial CAMs (the blue boxes) and successive localization maps obtained from images manipulated by iterative adversarial climbing, with the regularization procedure (*top*) and without (*bottom*).

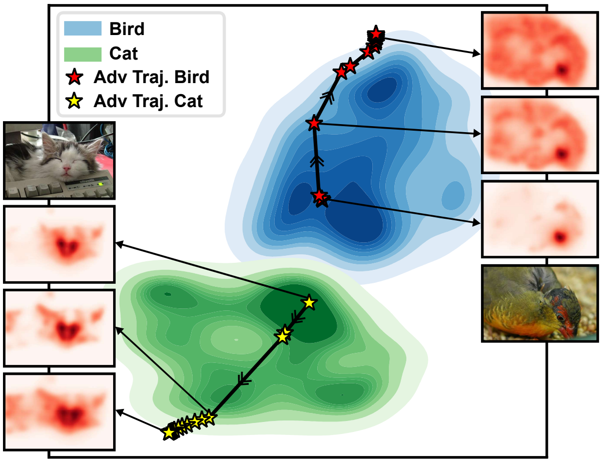


Figure 3.9: Feature manifold of images with “bird” (blue) and “cat” (green), and a trajectory of adversarial climbing for an image of each class. The dimensionality of the feature was reduced by t-SNE [141].

Step Size ξ: It determines the extent of the manipulation to the image in Eq. 3.5. Figure 3.7(e) shows the mIoU of the initial seed for different values of ξ. In our system, changes in step size ξ are not particularly significant.

Generality of Our Method: In addition to IRN [2], we experimented with two state-of-the-art methods of generating an initial seed for weakly supervised semantic segmentation, namely Chang *et al.* [30] and SEAM [107]. We used the authors’ pre-trained classifier where possible, but we re-trained the classifier of IRN [2] since the authors do not provide pre-trained one. We also followed their experimental settings including the backbone networks and mask refinement methods, *i.e.,* we used PSA [29] to refine the initial seed from “Chang *et al.* + AdvCAM” or “SEAM + AdvCAM”. Table 3.4 gives mIoU values for the initial seed and the pseudo ground truth mask obtained by combining each method with adversarial climbing. The use of AdvCAM improves the quality of the initial seed by an average of over 4%. Our approach does not require those initial seed generators to be modified or retrained.

Manifold Visualization: For visualizing a trajectory of adversarial climbing at a feature-level, we used t-SNE dimensional reduction [141]. We collect images that contain a single class of a cat or a bird and that are predicted by the classifier correctly. We then construct a set $\mathcal{F}$ containing the features of those images, before the final classification layer. We also choose a representative image of a cat, and another of a bird, and construct a set $\mathcal{F}'$ containing the features of those two images and their 20 manipulated images by adversarial climbing. Figure 3.9 presents t-SNE visualization of features in $\mathcal{F} \cup \mathcal{F}'$. We can see that adversarial climbing actually pushes the features away from the decision boundary boundary that separates the blue and green areas. In addition, despite 20 adversarial climbing steps, the manipulated features did not deviate significantly from the feature manifold of each class.

3.3.5 Analysis of Results by Class

The objects in the images in the MS COCO 2014 dataset are of various classes with various object sizes. We will now discuss the degree of improvement in the initial seed for each object class. Fig. 3.10 shows the improvement in mIoU produced by adversarial climbing over the initial seed for each class. The classes are listed in ascending order according to the average size of the target objects in each class (smallest → largest). Adversarial climbing improves mIoU values for the majority of classes, regardless of their average object size. When considering specific classes, we observed a large drop in the seed quality for the 'dining table' class, which is anomalous. We believe that this is due to the ambiguity of the ground truth label of the 'dining table'. In the MS COCO 2014 dataset, the 'dining table' label includes all the items on the table. The suppression of other classes by the regularization prevents objects such as bowls and bottles on the table from being identified as part of a 'dining table', resulting in a localization map that does not entirely match the ground truth.

To take a closer look at how adversarial climbing affects the performance of each class with various object sizes, we report precision, recall, and F1-score values averaged across all classes, the classes corresponding to the 10 smallest objects, and the classes

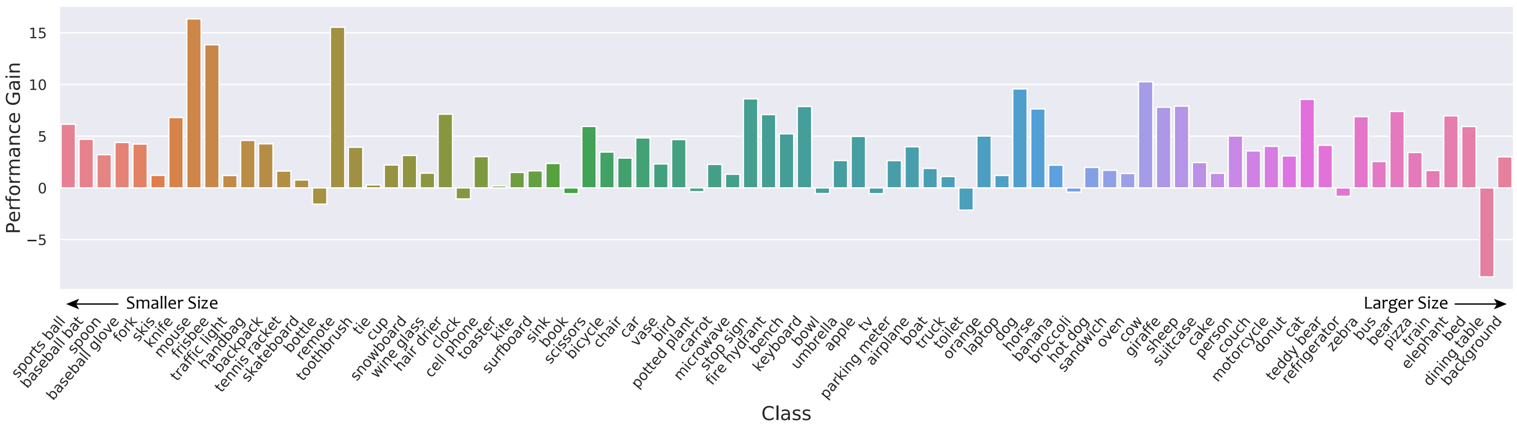


Figure 3.10: Per-class seed quality improvement in mIoU (%p) of IRN [2] achieved by adversarial climbing on the MS COCO 2014 training images. Classes are sorted by their average sizes. These sizes are due to Choe *et al.* [1].

Table 3.5: Precision, recall, and F1-score averaged across all classes, the smallest 10 classes, and the largest 10 classes. All the results are computed for 6% of the images from the MS COCO 2014 dataset ($\approx$ 5000 images). The average sizes for each class were borrowed from the work of Choe *et al.* [1].

	All Classes			Smallest 10 classes			Largest 10 classes		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
CAM	44.5	61.6	47.6	11.5	**73.0**	19.2	69.3	60.4	63.5
AdvCAM (T=10, τ=0.5)	46.7	63.8	50.6	14.8	69.7	23.8	70.9	63.9	65.6
AdvCAM (T=20, τ=0.5)	47.1	**64.7**	51.3	16.5	67.4	25.4	70.9	66.0	66.5
AdvCAM (T=30, τ=0.5)	**48.1**	63.6	**51.6**	18.5	64.9	27.2	71.4	65.7	**66.7**
AdvCAM (T=30, τ=0.4)	48.0	63.6	51.4	17.1	66.6	26.1	**71.5**	64.9	66.3
AdvCAM (T=30, τ=0.6)	47.7	63.7	51.4	**19.2**	64.3	**28.1**	71.2	**66.1**	66.5

corresponding to the 10 largest objects in Table 3.5. Our method improves precision, recall, and F1-score of the initial seed, averaged across all classes. Recall was slightly reduced (-12%) for the classes corresponding to the 10 smallest objects, but precision increased significantly (67%), resulting in a largely improved F1-score. This indicates that, for small objects, adversarial climbing effectively suppresses unwanted high attribution scores in the background.

We believe that there are two causes of these improved results on small objects: 1) During adversarial climbing, the logits associated with classes other than the target are reduced, as described in Section 4.3, and thus patterns which are irrelevant to the target class are effectively suppressed; and 2) since adversarial climbing increases the scores of regions relevant to the target class, the scores of background regions are suppressed due to normalization.

Adversarial climbing improves both precision and recall for large objects, but recall increases by a much larger margin. This indicates that adversarial climbing effectively raises the attribution scores of regions of target objects that had not previously been identified.

We will now look at how the hyper-parameters interact with the object size. Table 3.5 shows the precision, recall, and F1-score values obtained using different values of T and τ. Across all classes, neither T nor τ had a significant influence, which accords with the results presented in Section 3.3.4. Looking at the 10 classes containing the largest target objects, we see a similar picture. However, the 10 classes containing the smallest objects seem to be a little more sensitive to the values of the hyper-parameters, but not sufficiently to be a cause for concern.

3.4 Reducing Information Bottleneck

3.4.1 Information Bottleneck

Given two random variables X and Y, the mutual information $\mathcal{I}(X;Y)$ quantifies the mutual dependence between the two variables. Data processing inequality (DPI) [142] infers that any three variables X, Y, and Z that form a Markov Chain $X \rightarrow Y \rightarrow Z$ satisfy $\mathcal{I}(X;Y) \geq \mathcal{I}(X;Z)$. Each layer in a DNN processes the input only from the previous layer, which means that the DNN layers form a Markov chain. Therefore, the information flow through these layers can be represented using DPI. More specifically, when an $L-$layered DNN generates an output $\hat{Y}$ from a given input X through intermediate features T_l $(1 \leq l \leq L)$, it forms a Markov Chain $X \rightarrow T_1 \rightarrow \cdots \rightarrow T_L \rightarrow \hat{Y}$, and the corresponding DPI chain can be expressed as follows:

$$\mathcal{I}(X;T_1) \geq \mathcal{I}(X;T_2) \geq \cdots \geq \mathcal{I}(X;T_{L-1}) \geq \mathcal{I}(X;T_L) \geq \mathcal{I}(X;\hat{Y}). \tag{3.7}$$

This implies that the information regarding the input X is compressed as it passes through the layers of the DNN.

Training a classification network can be interpreted as extracting maximally compressed features of the input that preserve as much information as possible for classification; such features are commonly referred to as minimum sufficient features (*i.e.,* discriminative information). The minimum sufficient features (optimal representations T^{*}) can be obtained by the *information bottleneck* trade-off between the mutual information of X and T (compression), and that of T and Y (classification) [40, 44]. In other words, $T^{*} = \mathrm{argmin}_T\ \mathcal{I}(X;T) - \beta\mathcal{I}(T;Y)$, where $\beta \geq 0$ is a Lagrange multiplier.

Shwartz-Ziv *et al.* [41] observe a *compression phase* in the process of finding the optimal representation T^{*}: when observing $\mathcal{I}(X, T_l)$ for a fixed l, $\mathcal{I}(X, T_l)$ steadily increases during the first few epochs, but decreases in the later epochs. Saxe *et al.* [42] argue that the compression phase is mainly observed in DNNs equipped with double-sided saturating non-linearities (*e.g.,* tanh and sigmoid), and is not observed in those equipped with single-sided saturating non-linearities (*e.g.,* ReLU). This implies that DNNs with single-sided saturating non-linearities experience less information bottleneck than those with double-sided saturating non-linearities. This can also be understood in terms of gradient saturation in the double-sided saturating non-linearities: the gradient of those non-linearities with respect to an input above a certain value saturates close to zero [143]. Therefore, features above a certain value will have near-zero gradients during the back-propagation process and be restricted from additionally contributing to the classification.

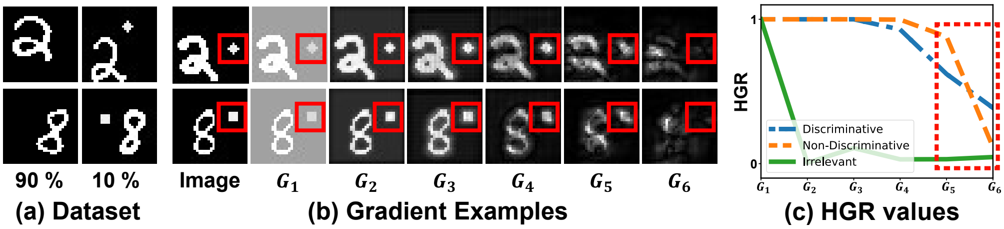
3.4.2 Motivation

As mentioned earlier, the DNN layers with double-sided saturating non-linearities have a larger information bottleneck than those with single-sided saturating non-linearities. The intermediate layers of popular DNN architectures (*e.g.,* ResNet [135] and DenseNet [144]) are coupled with the ReLU activation function, which is a single-sided saturating non-linearity. However, the final layer of these networks is activated by a double-sided saturating non-linearity such as sigmoid or softmax, and the class probability p is computed with the final feature map T_L and the final classification layer w, *i.e.,* $p = \texttt{sigmoid}(w^\intercal \text{GAP}(T_L))$. Therefore, the final layer parameterized by w has a significant bottleneck, and the amount of information transmitted from the last feature T_L to the actual classification prediction will be limited.

These arguments are analogous to the observations in existing methods. The information plane provided by Saxe *et al.* [42] shows that the compression of information is more noticeable in the final layer than in the other layers.

Bae *et al.* [109] observe that although the final feature map of the classifier contains rich information on the target object, the final classification layer filters out most of it; thus, the CAM cannot identify the entire area of the target object. This observation empirically supports the occurrence of the information bottleneck in the final layer of a DNN.

To take a closer look at this phenomenon, we design a toy experiment. We collect



(a) Dataset **(b) Gradient Examples** **(c) HGR values**

Figure 3.11: (a) Examples of toy images. (b) Examples of gradient maps G_k. (c) Plot of HGR values of $\mathcal{R}_{\text{D}}$, $\mathcal{R}_{\text{ND}}$, and $\mathcal{R}_{\text{BG}}$ for each layer, averaged over 100 images.

images containing the digits '2' or '8' from the MNIST dataset [145]. For only a small subset (10%) of these images, we add a circle ([black, fill=black]3pt) and a square (■) to the images containing the digits '2' and '8', respectively, at a random location (see Figure 3.11(a)). When classifying images into the digits '2' or '8', pixels corresponding to the digit are discriminative regions ($\mathcal{R}_{\text{D}}$), those corresponding to the added circle or square are non-discriminative but class-relevant regions ($\mathcal{R}_{\text{ND}}$), and those corresponding to the background are class-irrelevant regions ($\mathcal{R}_{\text{BG}}$).

We train a neural network with five convolutional layers followed by a final fully connected layer. We obtain the gradient map G_l of each feature T_l with respect to an input image x: $G_l = \nabla_x \sum_{u,v} T_l(u, v)$, where u and v are the spatial and channel indices of the feature T_l, and for the final classification layer ($l = 6$), $G_6 = \nabla_x y^c$. Because this gradient map indicates the extent to which each pixel of the image affects each feature, it can be used to examine how much information is passed from the input image to the feature maps of successive convolution layers.

We present examples of G_l in Figure 3.11(b). As an input image passes through the convolution layers, the overall amount of gradient with respect to the input decreases, indicating the occurrence of the information bottleneck. Specifically, the gradient of $\mathcal{R}_{\text{BG}}$ decreases early on ($G_1 \rightarrow G_2$), which implies that the task-irrelevant information is rapidly compressed. From G_1 to G_5, the gradient in $\mathcal{R}_{\text{D}}$ or $\mathcal{R}_{\text{ND}}$ gradually decreases. However, the decrease in the amount of gradient is prominent in the final layer ($G_5 \rightarrow G_6$), and in particular, the gradients in $\mathcal{R}_{\text{ND}}$ (red boxes) almost disappear. This supports our argument that there is significant information bottleneck in the final layer of a DNN, while also highlighting that the non-discriminative information in $\mathcal{R}_{\text{ND}}$ is particularly compressed.

We analyze this quantitatively. We define the high gradient ratio (HGR) of region $\mathcal{R}$ as the ratio of pixels that have a gradient above 0.3 to the total pixels in region $\mathcal{R}$. HGR quantifies the amount of transmitted information from region $\mathcal{R}$ of an input image to each feature. The trend in the HGR values of each region for each layer is shown in Figure 3.11(c). The observed trend is analogous to the above empirical observation, once again supporting that significant information bottleneck for $\mathcal{R}_{\text{ND}}$ occurs in the final layer (the red box).

We argue that the information bottleneck causes the localization map obtained from a trained classifier to focus on small regions of the target object. According to Eq. 2.1, the CAM only includes information that is processed by the final classification weight w_c. However, because only a subset of the information in the feature is passed through the final layer w_c due to the information bottleneck, leaving out most of the non-discriminative information, CAM cannot identify the non-discriminative regions of the target object. It is undesirable to use such CAMs to train a semantic segmentation network, for which the entire region of the target object should be identified. Therefore, we aim to bridge the gap between classification and localization by reducing the information bottleneck.

3.4.3 Proposed Method

In Section 3.4.2, we observed that the information contained in an input image is compressed particularly in the final layer of the DNN, due to the use of the double-sided saturating activation function therein. Therefore, we propose a method to reduce the information bottleneck of the final layer by simply removing the sigmoid or softmax activation function used in the final layer of the DNN. We focus on a multi-class multi-label classifier, which is the default setting for weakly supervised semantic segmentation. Suppose we are given an input image x and the corresponding one-hot class label $t = [t_1, \cdots, t_{\mathcal{C}}]$, where $t_c \in \{0, 1\}$ $(1 \leq c \leq \mathcal{C})$ is an indicator of a class c, and $\mathcal{C}$ is the set of all classes. While existing methods use the sigmoid binary cross-entropy (BCE) loss ($\mathcal{L}_{\text{BCE}}$) to train a multi-label classifier, our method replaces it with another loss function $\mathcal{L}_{\text{RIB}}$ that does not rely on the final sigmoid activation function:

$$\mathcal{L}_{\text{BCE}} = -\sum_{c=1}^{\mathcal{C}} t_c \log \texttt{sigmoid}(y^c) + (1 - t_c) \log(1 - \texttt{sigmoid}(y^c)),$$

$$\mathcal{L}_{\text{RIB}} = -\sum_{c=1}^{\mathcal{C}} t_c \min(m, y^c),$$

where m is a margin, and y^c is the classification logit of image x.

However, training a classifier with $\mathcal{L}_{\text{RIB}}$ from scratch causes instability in the training because the gradient cannot saturate (please see the Appendix). Therefore, we first train an initial classifier with $\mathcal{L}_{\text{BCE}}$ whose trained weights are denoted by θ_0, and for a given image x, we adapt the weights toward a bottleneck-free model of x. Specifically, we fine-tune the initial model using $\mathcal{L}_{\text{RIB}}$ computed from x and obtain a model parameterized by θ_k $(0 < k \leq K)$, where $\theta_k = \theta_{k-1} - \lambda \nabla_{\theta_{k-1}} \mathcal{L}_{\text{RIB}}$, and K and λ are respectively the total number of iterations and the learning rate for fine-tuning. We name this fine-tuning process RIB. Employing RIB reduces the information bottleneck for x, and we can obtain CAMs that identify more regions of the target object, including non-discriminative regions. We repeat the RIB process for all the training images to obtain the CAMs.

However, the model that is adapted to a given image x can be easily over-fitted to x. Therefore, to further stabilize the RIB process, we construct a batch of size B for RIB by sampling random $B - 1$ samples other than x at each RIB iteration. Note that for each iteration, $B - 1$ samples are randomly selected, while x is fixed.

Effectiveness of RIB: We demonstrate the effectiveness of RIB by applying it to the same classifier as that used for the toy experiments described in Section 3.4.2. Figure 3.12 presents (a) examples of G_6 and (b) the HGR values for $\mathcal{R}_{\text{D}}$, $\mathcal{R}_{\text{ND}}$, and $\mathcal{R}_{\text{BG}}$

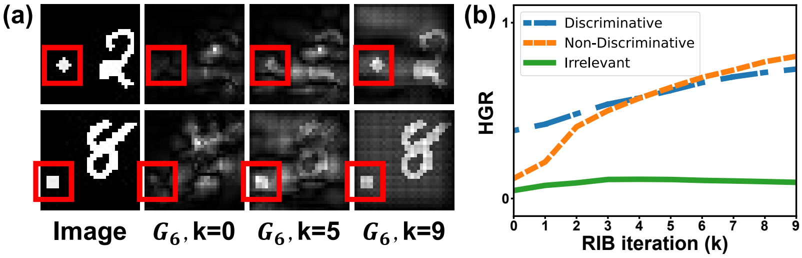


Figure 3.12: Analysis of G_6 for $\mathcal{R}_{\text{D}}$, $\mathcal{R}_{\text{ND}}$, and $\mathcal{R}_{\text{BG}}$ at each RIB iteration.

of G_6, which showed the most significant information bottleneck, at each RIB iteration. The HGR values are averaged over 100 images. The HGR values of $\mathcal{R}_{\text{BG}}$ remain fairly constant during the RIB process, while the HGR values of $\mathcal{R}_{\text{D}}$ and $\mathcal{R}_{\text{ND}}$ increase significantly. This indicates that the RIB process can indeed reduce the information bottleneck, thereby ensuring that more information corresponding to both $\mathcal{R}_{\text{D}}$ and $\mathcal{R}_{\text{ND}}$ is processed by the final classification layer.

Limiting the transmission of information from discriminative regions: Zhang *et al.* [89] showed the relationship between a classification logit y and a CAM, *i.e.*, $y = \text{GAP}(\texttt{CAM})$. This implies that increasing y^c with RIB also increases the pixel values in the CAM. For a CAM to identify a wider area of the target object, it is important to increase the pixel scores of the non-discriminative regions, rather than the discriminative regions. Therefore, we introduce a new pooling method to the RIB process, so that the features that were previously delivering a small amount of information to the classification logit contribute more to the classification.

We propose a global non-discriminative region pooling (GNDRP). Contrary to GAP which aggregates all the values of the spatial location in the feature map T_l, our GNDRP selectively aggregates the values of spatial locations whose CAM scores are below a threshold τ, as follows:

$$\text{GAP}(T_l) = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} T_l(u), \quad \text{GNDRP}(T_l) = \frac{1}{|\mathcal{U}_\tau|}\sum_{u\in\mathcal{U}_\tau} T_l(u),$$

$$\mathcal{U}_\tau = \{u \in \mathcal{U} \,|\, \texttt{CAM}(u) \leq \tau\},$$

where $\mathcal{U}$ is a set of all spatial location indices in T_l.

Other methods of weakly supervised semantic segmentation also considered new pooling methods other than GAP to obtain better localization maps [146, 119, 147]. The pooling methods introduced in previous works make the classifier focus more on discriminative parts. In contrast, GNDRP excludes highly activated regions, encouraging non-discriminative regions to be further activated.

Obtaining a final localization map: We obtain the final localization map $\mathcal{M}$ by aggregating all the CAMs obtained from the classifier at each RIB iteration k: $\mathcal{M} = \sum_{0 \leq k \leq K} \texttt{CAM}(x; \theta_k)$.

Weakly Supervised Semantic Segmentation: Because a CAM [36] is obtained from down-sampled intermediate features produced by a classifier, it should be up-sampled to the size of the original image. Therefore, it tends to localize the target object coarsely and cannot represent its exact boundary. Many weakly supervised semantic segmentation methods [30, 148, 107, 48, 131, 27] produce pseudo ground truths by modifying their initial seeds using established seed refinement methods [91, 29, 2, 119, 120]. Similarly, we obtain pseudo ground truths by applying IRN [2], a state-of-the-art seed refinement method, to the coarse map $\mathcal{M}$.

In addition, because an image-level class label is void of any prior regarding the shape of the target object, salient object mask supervision is popularly used in existing methods [52, 27, 90, 149]. Salient object mask supervision can also be applied to our method to refine the pseudo ground truths: when a foreground pixel in a pseudo label is identified as background on this map, or a background pixel is identified as foreground, we ignore such pixels in the training of the segmentation network.

3.4.4 Experiments

Dataset and evaluation metric: We evaluated our method quantitatively and qualitatively by conducting experiments on the PASCAL VOC 2012 [32] and the MS COCO 2014 [79] datasets. Following the common practice in weakly supervised semantic segmentation [29, 2, 27, 48], we used the PASCAL VOC 2012 dataset, which is augmented by Hariharan *et al.* [133], containing 10,582 training images with objects from 20 classes. The MS COCO 2014 dataset contains approximately 82K training images containing objects of 80 classes. We evaluated our method on 1,449 validation images and 1,456 test images from the PASCAL VOC 2012 dataset and on 40,504 validation images from the MS COCO 2014 dataset, by calculating the mean intersection-over-union (mIoU) values.

Reproducibility. We implemented CAM [36] by following the procedure from Ahn *et al.* [2], which is implemented with the PyTorch framework [116]. We used the ResNet-50 [135] backbone for the classification. We fine-tuned our classifier for $K = 10$ iterations with a learning rate of 8×10^{-6} and a batch size of $B = 20$. We set the margin m to 600. For the GNDRP, we set τ to 0.4. For the final semantic segmentation, we used the PyTorch implementation of DeepLab-v2-ResNet101 offered by [150]. We used an initial model pre-trained on the ImageNet dataset [3]. For the MS COCO 2014 dataset, the training images are cropped with the crop size of 481×481 rather than 321×321 used for the PASCAL VOC 2012 dataset, considering the size of the images in this dataset.

Quality of the initial seed and pseudo ground truth on the PASCAL VOC 2012 dataset: In Table 4.2, we report the mIoU values of the initial seed and pseudo ground truth masks generated from our method and from other recent techniques. Following SEAM [107], we evaluate a range of thresholds to distinguish between the foreground and the background in the map $\mathcal{M}$ and then determine the best initial seeds. Our initial seeds exhibit 7.7%p improvement from the original CAMs, a baseline for comparison, and simultaneously outperform those from the other methods. Note that our initial seeds are better than those of SEAM, which further refines the initial CAM on a pixel-level by considering the relationship between pixels through an auxiliary self-attention module.

We applied a post-processing method based on conditional random field (CRF) [75] for pixel-level refinement of the initial seeds obtained from the method proposed by Chang *et al.* [30], SEAM [107], IRN [2], and our method. On average, applying CRF improved all the seeds by more than 5%p, with the exception of SEAM. CRF improved SEAM by only 1.4%p, and it is reasonable to believe that this unusually small improvement occurred because the self-attention module had already refined the seed from CAM. When the seed produced by our method is refined with CRF, it is 6.1%p better than that from SEAM and consequently outperforms all the recent competitive methods by a large margin.

Additionally, we compare the pseudo ground truth masks obtained after seed refinement with those obtained using other methods. Most of the compared methods use PSA [29] or IRN [2] to refine their initial seeds. For a fair comparison, we generate pseudo ground truth masks using both seed refinement techniques. Table 4.2 shows that the masks from our method yield an mIoU of 68.6 with PSA [29] and 70.6 with IRN [2], thereby outperforming other methods by a large margin.

Quality of the initial seed and pseudo ground truth on the MS COCO 2014 dataset: Table 4.2 presents the mIoU values of the initial seed and pseudo ground truth masks obtained by our method and by other recent methods for the MS COCO 2014 dataset. We obtained the results of IRN [2] using the official code to set the baseline performance. Our method improved the initial seed and pseudo ground truth masks of our baseline IRN [2], by mIoU margins of 3.0%p and 2.7%p, respectively.

Figure 3.13 illustrates localization maps gradually refined by the RIB process for the PASCAL VOC 2012 and the MS COCO 2014 datasets. More samples are shown in the Appendix.

Weakly supervised semantic segmentation performance on the PASCAL VOC 2012 dataset: Table 5.3 presents the mIoU values of the segmentation maps on PASCAL VOC 2012 validation and test images, predicted by our method and other recently introduced weakly supervised semantic segmentation methods, which use bounding box labels or image-level class labels. All the results in Table 5.3 were obtained using a ResNet-based backbone [135]. Our method achieves mIoU values of

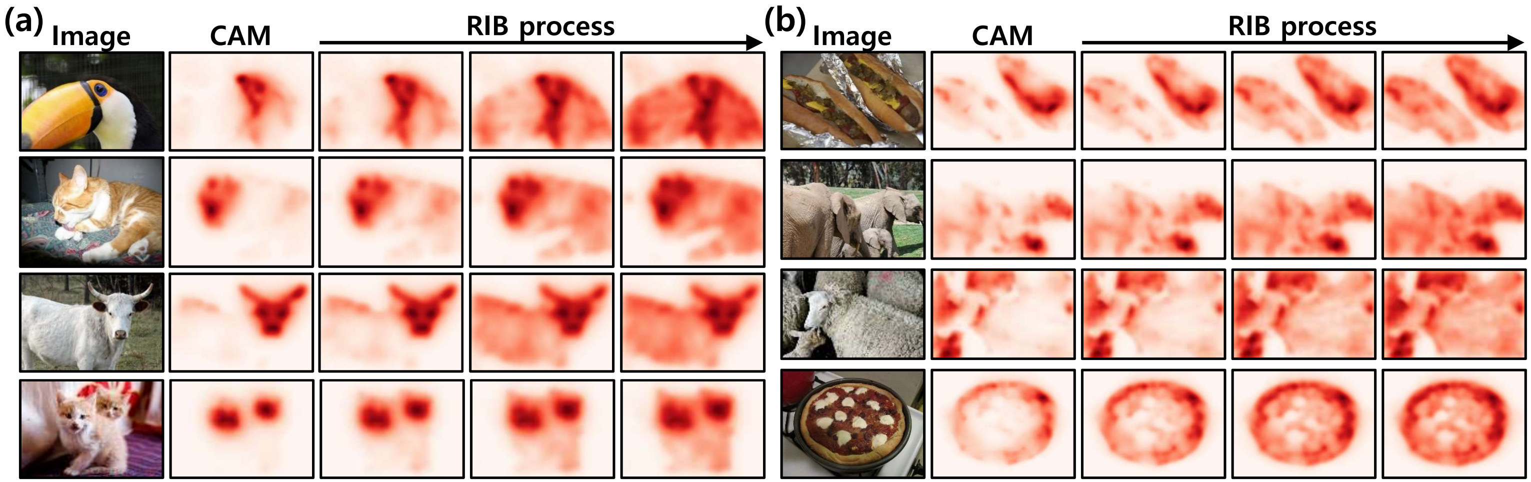


Figure 3.13: Examples of localization maps obtained during the RIB process for (a) PASCAL VOC 2012 training images and (b) MS COCO 2014 training images.

68.3 and 68.6 for the validation and test images, respectively, on the PASCAL VOC 2012 semantic segmentation benchmark, outperforming all the methods that use image-level class labels as weak supervision. In particular, our method outperforms CONTA [48], the best-performing method among our competitors, achieving an mIoU value of 66.1. However, CONTA depends on SEAM [107], which is known to outperform IRN [2]. When CONTA was implemented with IRN for a fairer comparison with our method, its mIoU value decreased to 65.3, which our method surpasses by 3.0%p.

Table 3.7 compares our method with other recent methods using additional salient object supervision. We utilized salient object supervision used by Li *et al.* [149] and Yao *et al.* [52]. Our method achieves mIoU values of 70.2 and 70.0 for the validation and test images, respectively, outperforming all the recently introduced methods under the same level of supervision.

Figure 3.14(a) shows examples of predicted segmentation maps by our method with and without saliency supervision. The boundary information provided by saliency supervision allows our method to produce a more precise boundary (yellow boxes). However, the non-salient objects in an image are often ignored when using saliency supervision, while RIB successfully identifies them (*e.g.*, a 'sofa' in the first column and 'person' in red boxes in Figure 3.14(a)). This empirical finding inspires a potential future work that can simultaneously identify a precise boundary and non-salient objects.

Performance of weakly supervised semantic segmentation on the MS COCO 2014 dataset: Table 3.8 compares our method with other recent methods on MS COCO 2014 validation images. Our method achieves an improvement of 2.4%p in terms of the mIoU score compared with our baseline IRN [2], and outperforms the other recent competitive methods [1, 48, 153] by a large margin. In the comparison with CONTA [48], the result of IRN reported in CONTA [48] differs from the one we obtained. Therefore, we compare relative improvements: CONTA achieves a 0.8%p improvement compared with IRN (32.6 $\rightarrow$ 33.4), whereas our method achieves 2.4%p (41.4 $\rightarrow$ 43.8). Figure 3.14(b) presents examples of predicted segmentation maps by

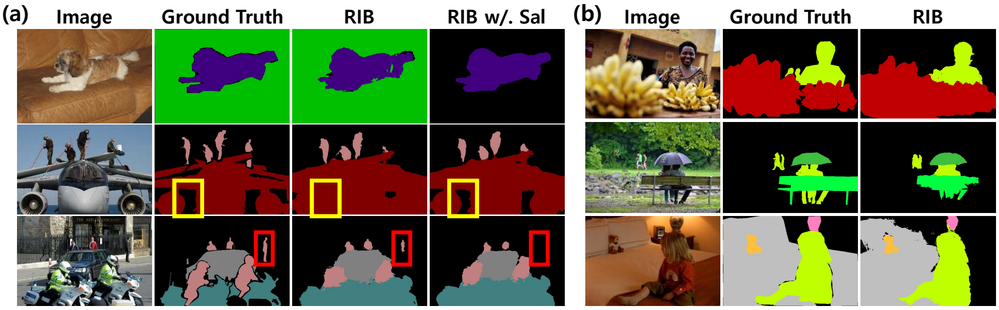


Figure 3.14: Examples of predicted segmentation masks from IRN [2] and our method for (a) PASCAL VOC 2012 validation images and (b) MS COCO 2014 validation images.

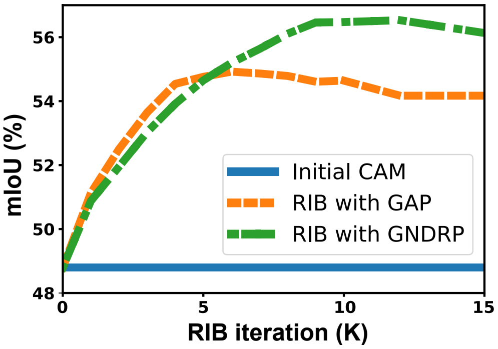


Figure 3.15: Analysis of RIB with GAP or GNDRP in terms of mIoU of the initial seed.

our method for the MS COCO 2014 validation images.

Influence of the total number of RIB iterations K**:** We analyze the influence of the iteration number K on the effectiveness the RIB process. Figure 3.15 shows the mIoU score of the initial seed obtained by our baseline CAM, and that of each iteration of the RIB process with GAP or GNDRP. As the RIB process progresses, the localization map is significantly improved, regardless of the pooling method. However, the increase in the performance of RIB with GAP is limited, and even slightly decreases in later iterations ($K > 5$). This is because GAP allows features that have already delivered sufficient information to the classification to become even more involved in

Method	*val*	*test*
Supervision: Bounding box labels		
Song *et al.* $_{\text{CVPR '19}}$ [25]	70.2	-
BBAM $_{\text{CVPR '21}}$ [26]	73.7	73.7
Supervision: Image class labels		
IRN $_{\text{CVPR '19}}$ [2]	63.5	64.8
SEAM $_{\text{CVPR '20}}$ [107]	64.5	65.7
BES $_{\text{ECCV '20}}$ [120]	65.7	66.6
Chang *et al.* $_{\text{CVPR '20}}$ [30]	66.1	65.9
RRM $_{\text{AAAI '20}}$ [151]	66.3	66.5
CONTA $_{\text{NeurIPS '20}}$ [48]	66.1	66.7
RIB (Ours)	**68.3**	**68.6**

Table 3.6: Comparison of semantic segmentation performance on PASCAL VOC 2012 validation and test images.

Method	Sup.	*val*	*test*
SeeNet $_{\text{NeurIPS '18}}$ [90]	$\mathcal{S}$	63.1	62.8
FickleNet $_{\text{CVPR '19}}$ [27]	$\mathcal{S}$	64.9	65.3
CIAN $_{\text{AAAI '20}}$ [118]	$\mathcal{S}$	64.3	65.3
Zhang *et al.* $_{\text{ECCV '20}}$ [138]	$\mathcal{S}$	66.6	66.7
Fan *et al.* $_{\text{ECCV '20}}$ [152]	$\mathcal{S}$	67.2	66.7
Sun *et al.* $_{\text{ECCV '20}}$ [94]	$\mathcal{S}$	66.2	66.9
LIID $_{\text{TPAMI '20}}$ [95]	$\mathcal{S}_I$	66.5	67.5
Li *et al.* $_{\text{AAAI '21}}$ [149]	$\mathcal{S}$	68.2	68.5
Yao *et al.* $_{\text{CVPR '21}}$ [52]	$\mathcal{S}$	68.3	68.5
RIB (Ours)	$\mathcal{S}$	**70.2**	**70.0**

Table 3.7: Comparison of semantic segmentation performance on PASCAL VOC 2012 validation and test images using explicit localization cues. $\mathcal{S}$: salient object, $\mathcal{S}_I$: salient instance.

the classification. Because our proposed GNDRP limits the increase in the contribution of these discriminative regions to the classification, RIB with GNDRP can effectively allow non-discriminative information to be more involved in the classification, resulting in a better localization map in later iterations. We observe that changing the value of K to be larger than 10 (even 20) produces less than 0.8%p drop in mIoU, suggesting that it is not difficult to select a good value of K.

Fine-tuning with $\mathcal{L}_{\text{RIB}}$: To verify the effectiveness of $\mathcal{L}_{\text{RIB}}$, we fine-tune a model using the BCE loss with various double-sided saturating activation functions. Table 3.9 (a) shows the mIoU scores of the initial seeds, obtained from a model fine-tuned by the BCE loss with sigmoid, tanh, and softsign activations, and our $\mathcal{L}_{\text{RIB}}$. We adjusted the output of tanh and softsign to have a value between zero and one through the

Method	Backbone	mIoU
ADL $_{\text{TPAMI '20}}$ [1]	VGG16	30.8
CONTA $_{\text{NeurIPS '20}}$ [48]	ResNet50	33.4
Yao *et al.* $_{\text{Access '20}}$ [153]	VGG16	33.6
IRN $_{\text{CVPR '19}}$ [2]	ResNet101	41.4
RIB (Ours)	ResNet101	43.8

Table 3.8: Comparison of semantic segmentation on MS COCO validation images.

Fine-tuning	Seed
Init.	48.8
BCE w/. Tanh	49.7
BCE w/. Sigmoid	50.5
BCE w/. Softsign	50.9
$\mathcal{L}_{\text{RIB}}$	**56.5**

(a)

m	λ	Seed
300	8×10^{-6}	54.0
600	5×10^{-6}	54.9
600	8×10^{-6}	**56.5**
600	1×10^{-5}	56.0
1000	8×10^{-6}	55.9

(b)

Method	Seed
CAM	48.8
RIB-GAP	54.8
RIB-GNDRP (τ=0.3)	55.8
RIB-GNDRP (τ=0.4)	**56.5**
RIB-GNDRP (τ=0.5)	56.0

(c)

Table 3.9: Comparison of mIoU scores of the initial seed (a) with different activation functions for the final layer, (b) with different values of m and λ, and (c) with different values of τ.

affine transform. Fine-tuning using the BCE loss with double-sided saturating activations improves the initial seed to some extent, which demonstrates the effectiveness of per-sample adaptation; however, their performance improvement is limited due to the remaining information bottleneck. Note that the softsign activation function provides better localization maps than tanh and sigmoid. We believe this is because the gradients from softsign reach zero at a higher value compared with the others (please see the Appendix), and consequently, softsign has less information bottleneck. Our $\mathcal{L}_{\text{RIB}}$ effectively addresses the information bottleneck and achieves the best performance.

Analysis of the sensitivity to hyper-parameters: We analyze the sensitivity of the mIoU of the initial seed to the hyper-parameters involved in the RIB process. Table 3.9 (b) presents the mIoU values of the initial seed obtained using different combinations of values for the margin m and the learning rate λ. Overall, a slightly lower performance is observed when the strength of the RIB process is weakened by small values of m and λ. For sufficiently large m and λ, the performance of the RIB process is competitive. Table 3.9 (c) analyzes the influence of the threshold τ involved in the GNDRP. Increasing τ from 0.3 to 0.5 results in less than 1%p change in the mIoU, and thus, we conclude that the RIB process is robust against the changes in τ.

3.5 Summary

In this chapter, we have introduced three weakly supervised semantic segmentation methods using image-level class labels. To expand the regions of CAMs to the extent of the target object, we have first briefly discussed FickleNet, the stochastic inference technique. We have shown how adversarial manipulation can be used to expand the small discriminative regions of a target object. We manipulate images with a pixel-level perturbation, which is obtained from the gradient computed from the output of classifier with respect to the input image, which increase the classification score of the perturbed image. The attribution map of the manipulated image covers more of the target object. Finally, we analyzed why the localization map obtained from a classifier identifies only a small region of the target object through the information bottleneck principle. Our analysis highlighted that the amount of information delivered from an input image to the output classification is largely determined by the final layer of the DNN. We then developed a method to reduce the information bottleneck through two simple modifications to the existing training scheme: the removal of the final non-linear activation function in the DNN and the introduction of a new pooling method. We have shown that these techniques are helpful for obtaining improved localization maps, which identify accurate regions of the target object.

Chapter 4

Learning with Auxiliary Data

4.1 Introduction

Pixel-wise labeling is labor-intensive [22]. Lots of research have been dedicated to supervising a semantic segmentation model with weaker forms of supervision than pixel-wise labelings, such as scribbles [23], points [21, 154], boxes [24, 25, 26], and class labels [107, 112, 49, 59]. We tackle the final category in this paper: weakly supervised semantic segmentation (WSSS) with class labels.

WSSS methods utilizing class labels often follow a two-stage process. First, they generate pixel-level pseudo-target from a classifier using CAM variants [36, 37]. Then, they train the main segmentation network using the pseudo-target generated in the first stage. Built on image-level labels only, the pseudo-target is known to suffer from the confusion between foreground and background cues. For example, given a database of duck images where ducks are typically waterborne, a classifier erroneously assigns higher scores on patches containing water than those containing ducks' feet [46, 47, 48, 49, 50, 51]. The same goes for foreground-background pairs like woodpecker-tree, snowmobile-snow, and train-rail. This is a fundamental problem that cannot be solved solely with the class labels; additional information is needed to learn to fully distinguish the foreground and background cues [46, 47, 50].

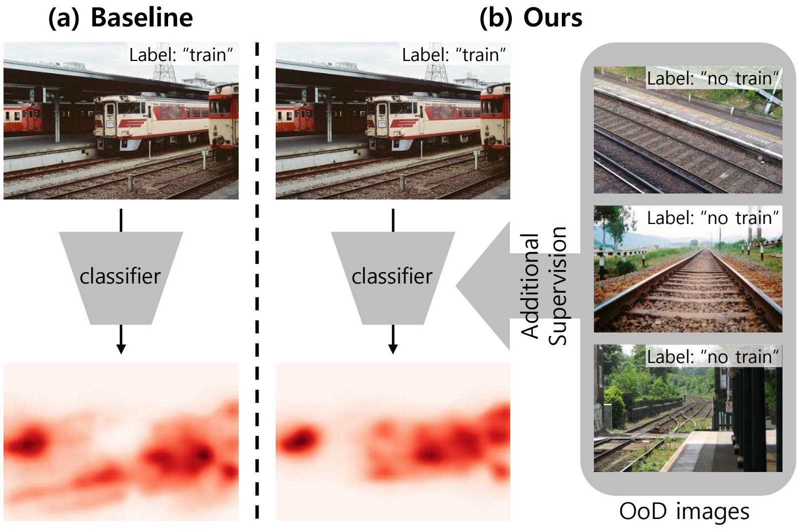


Figure 4.1: (a) Classifiers often confuse background cues to be a foreground concept due to spurious correlations ("rail" for "train"). (b) Our W-OoD employs hard OoD images as negative samples ("rail" is not "train") to resolve the confusion.

Researchers have thus sought various sources of additional guidance to separate the foreground and background cues, each with different pros and cons and different labeling-cost footprints. Image saliency [92, 93] is one of the most widely used ones [27, 50, 52, 94, 95, 96], for it naturally provides the prominent foreground object in the image in a class-agnostic fashion. However, saliency is not very effective for non-salient foreground objects (low-contrast objects or small objects). Low-level visual features like superpixels [53, 54], edges [97], object proposals [26, 25, 95], and optical flows [34, 33] have also been considered. Though cost-effective, they tend to generate inaccurate object boundaries because such low-level information does not consider semantics associated with the class.

In this paper, we propose another source of guidance that provides a distinction between the foreground and background cues. We propose to use the *out-of-distribution (OoD)* data that do not contain any of the foreground classes of interest. Examples include the rail-only images for the foreground class "train", since classifiers often confuse the rail for the train. By subduing the recognition of “train” on such rail cues in hard OoDs, models successfully distinguish such confusing cues.

Obtaining such OoDs does not incur a significant amount of additional annotation efforts compared to collecting only the image-level labels. The OoD images are natural by-products of the typical dataset collection procedure. Vision datasets with image-level category labels (Pascal [32], COCO [79], LVIS [155], and OpenImages [156]) all start with a pool of candidate images, from which images corresponding to one of the foreground classes are selected and included in the final dataset. The remaining pool, or the *candidate OoD set*, can be utilized as the source of OoD images.

The candidate OoD set cannot be directly used for guiding the WSSS method for two reasons. First, general OoD images do not provide informative signals to distinguish difficult background cues from the foreground (rail from train). Second, it may still contain foreground objects. We address the first problem by selecting *hard OoDs* whereby classifiers falsely assign high prediction scores to one of the foreground classes. The second problem is addressed by a human-in-the-loop process where images containing foreground objects are manually pruned. While this requires additional human efforts, we emphasize that the extra cost is negligible. As we will show later (Sec. 4.4.3), we only need a tiny amount of hard OoD samples to improve the localization maps: even 1 hard OoD image per class boosts the localization performance by 2.0%p. Furthermore, the cost for collecting OoD samples is at the same order of magnitude as collecting the category labels for the foreground samples, as opposed to collecting segmentation maps. One can also re-direct the budget for collecting a few labeled foreground data to collecting a similar number of hard OoD samples to dramatically improve the WSSS performance.

Given the additional guidance provided by OoD samples, we propose **W-OoD**, a method of training a classifier by utilizing the hard-OoDs. Note that our data collection procedure provides hard OoD samples which have different patterns and semantics in various contexts. One could ignore this diversity and treat every hard OoD as a combined background class; this approach has proved to be sub-optimal by our experiments. Instead, W-OoD considers every hard OoD sample with a metric-learning objective: increase the distance between the in-distribution and OoD samples in the feature space. This forces the background cues shared by the in-distribution and OoD samples (rail for train category) to be excluded from the feature-space representation. W-OoD results in high-quality localization maps and lead to the new state-of-the-art performance on the Pascal VOC 2012 benchmark for WSSS.

We contribute (1) a new paradigm of utilizing the OoD samples to address the spurious correlations in weakly supervised semantic segmentation (WSSS); (2) a dataset of hard OoDs for 20 Pascal categories that will be published upon acceptance; and (3) a WSSS method, W-OoD, that exploits the hard OoDs and achieves the best-known performance on the Pascal VOC 2012 benchmark for WSSS.

4.2 Related Work

Weakly supervised learning: Most weakly supervised learning methods with image-level class labels are based on a class activation map (CAM) [36]. However, it is widely known that a CAM is limited to identifying small discriminative parts of a target object [27, 2, 49]. Several techniques have been proposed for obtaining the entire region of the target object. PSA [29] and IRN [2] consider pixel relationships to extend the object region to semantically similar areas using a random walk. SEAM [107] regularizes the classifier so that the localization maps obtained from differently transformed images are equivariant to those transformations. AdvCAM [112] and RIB [49] propose post-processing techniques of a trained classifier to obtain whole regions of the target object, by manipulating images or network weights. Although the identified regions are successfully extended by these methods, some spuriously correlated background regions tend to be erroneously identified together. CDA [157] adopts the cut-paste method to decouple the correlation between objects and their contextual background.

However, it is difficult to accurately decouple the correlation using only class labels, which limits the performance improvement.

Learning with external data: Several studies have considered utilizing additional external information to address the issue of the spurious correlation problem. Automated web searches can provide images [35, 158] or videos [34, 33] with class labels, although these labels may be inaccurate. Some methods [137, 94] utilize single-label images to obtain more information about in-distribution data. However, these additional sources still depend solely on classes of interest. Thus, they lack information about the separation between the foreground and background. Consequently, various types of additional supervision have been adopted. Some researchers [159, 160] employed image captions. However, these are expensive to obtain. Moreover, modeling vision-language relationships, which is required in those methods, is a non-trivial task. Kolesnikov *et al.* [51] proposed an active learning approach, wherein a person determines whether a specific pattern is in the foreground or not. This is a model-specific approach, so human intervention is required whenever a new model is trained. Saliency supervision [161, 162] is another popular additional information source [49, 33, 94, 163, 52, 50, 96]. However, it is not very effective for non-salient objects that are indistinguishable from the background or small objects [49, 163, 50].

4.3 Methods

We propose a method for collecting and utilizing OoD data for the WSSS with category labels. We describe the data collection procedure for hard OoD in Sec. 4.3.1. In Sec. 4.3.2, we introduce the method named W-OoD that trains a classifier with the collected hard-OoDs to generate the localization maps. Finally in Sec. 4.3.3, we show how to train a semantic segmentation network with the localization maps.

4.3.1 Collecting the Hard Out-of-Distribution Data

We describe the overall procedure for collecting an OoD dataset. The starting point is a *candidate OoD set* that consists mostly of images without the foreground categories of interest. The aim is to refine this set into a set of hard OoDs that will be used for the downstream WSSS methods. The overall procedure is described in Fig. 4.2.

Where to get the candidate OoDs: The WSSS task with category labels as the weak supervision first requires the category labels on a set of training images. Building a category-labeled image dataset is typically a four-step process [32, 156, 79, 155]: (1) define the list $\mathcal{C}$ of foreground classes of interest, (2) acquire unlabelled images from various sources (world wide web), (3) determine for each image whether it contains one of the foreground classes, and (4) tag each image with the foreground category labels. Steps (3) and (4) are combined in some cases. A by-product of this procedure is the set of candidate images obtained from step (2) but not selected in step (3). We refer to this set as the *candidate OoD set*. For example, for Pascal VOC 2007 [32], step (2) has yielded 44,269 candidate images for annotation. Everingham *et al.* [32] report that 9,963 of them were finally selected as foreground data, while the rest were discarded. We make use of this discarded set that is likely to consist of background images.

Hard OoD samples via ranking and pruning: Unfortunately, the candidate OoD data are imperfect. OoD data are often too diverse to contain meaningful information. For example, presenting an image of fish in an aquarium as a negative sample of the foreground class "train" will not introduce any meaningful supervision for the classifier (See fish in Fig. 4.2). It is the *hard OoD samples* that give much information; they are OoD samples confused by a classifier to be containing the foreground object. The rail images *without train* in Fig. 4.2 are examples of such. They provide informative negative supervision for the classifier to suppress the class score on spurious background cues. We thus rank the candidate OoD data according to the prediction scores $p(c)$ for the class c of interest. We use the classifier trained on the images with foreground objects and the corresponding labels. We prune OoD samples with $p(c) < 0.5$. This

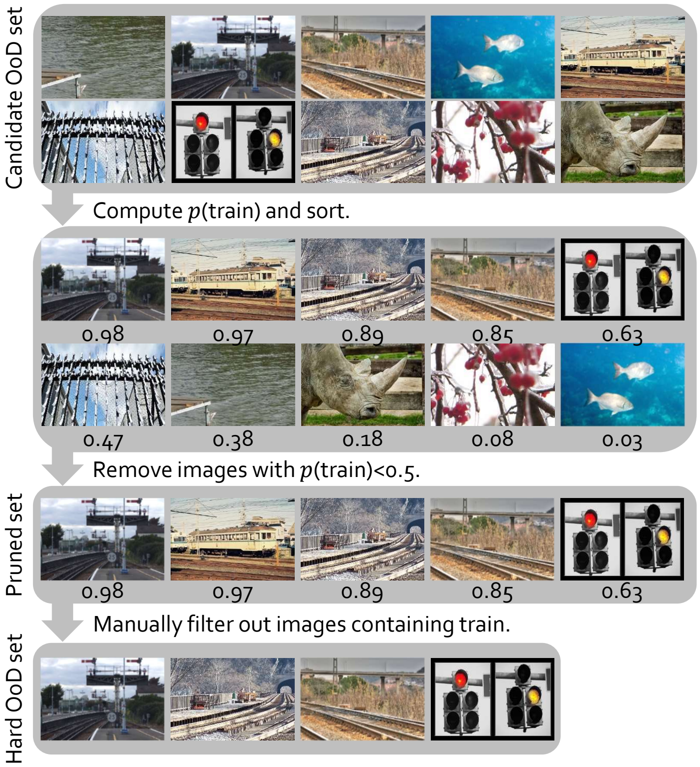

Candidate OoD set
Compute p(train) and sort.
0.98
0.97
0.89
0.85
0.63
0.47
0.38
0.18
0.08
0.03
Remove images with p(train)<0.5.
Pruned set
0.98
0.97
0.89
0.85
0.63
Manually filter out images containing train.
Hard OoD set

Figure 4.2: **Collecting hard OoD data**. Starting from the candidate OoD images at the top, we sequentially prune out easy OoDs and then false negatives for each foreground class $c \in \mathcal{C}$. The procedure results in the **hard OoD dataset**.

returns candidates for the hard OoD data.

Manual pruning of positive samples: It is unrealistic to assume that the candidate OoD set will be free of foreground objects. There will be many missing annotations and corner cases. When they are ranked according to the foreground prediction scores, high-ranking images are likely to contain those missing positives. We thus need to manually filter out those positive samples. This manual refinement stage is the cost bottleneck in our pipeline. The cost depends directly on the *positive rate* r, the proportion of positive images among the pruned set obtained by thresholding the prediction score $p(c) \geq 0.5$. Letting n be the required number of hard OoD images, the human worker needs to check on average $\frac{n}{1-r}$ images. If there are some positive images with $r = 0.2$, then the annotator needs to check $1.25n$ images to eventually obtain n hard OoDs. We denote the resulting dataset as $\mathcal{D}_{\text{ood}}$, the *hard OoD set*.

Surrogate source of OoD data: Theoretically speaking, it would be best to obtain the hard OoD set by replicating the dataset construction procedure for Pascal [32] to analyze and benchmark our method on Pascal. However, this is practically infeasible because one cannot crawl images with similar characteristics as the 500,000 initial images that Pascal authors have crawled from Flickr in 2007 [32]. It is also not documented which category annotation tool has been used to filter out the background set. Another way to set up the experiment is to build a new dataset from scratch. However, this will not allow us to use the existing WSSS benchmarks like Pascal. In this paper, we source the candidate OoD data from another vision dataset: OpenImages [156]. To simulate the OoD data, we filter out 20 Pascal classes from the OpenImages dataset using the provided category labels. Note that OpenImages category labels are noisy: 19,794 categories are labeled first through image classifiers and then are refined by crowdsourced workers [156]. This is in stark contrast to Pascal: only 20 categories are labeled by a highly controlled pool of workers at a controlled offline event (called "annotation party") [32]. We thus expect the candidate OoD set sourced from OpenImages to contain more noise (foreground classes) than the set one would get from the original Pascal data collection process.

4.3.2 Learning with the Hard Out-of-Distribution Data

Classifiers trained only on the in-distribution dataset $\mathcal{D}_{\text{in}}$ often incorrectly identify spuriously correlated background regions as class-relevant patterns. We address this by using the hard out-of-distribution data $\mathcal{D}_{\text{ood}}$ obtained in the previous section. One naive approach to utilize the hard OoD images is either to assign the uniform distribution over the labels for such images (no-information prior) [164, 165, 166] or to assign the "background" label to such images. However, since hard OoD images contain various semantics that convey meaningful information to each class, labeling these images with one background class ignores the diversity of OoD samples, resulting in a sub-optimal performance as shown in Sec. 4.4.3 and Table 4.5.

To benefit from the diversity of hard-OoD images, we propose a metric-learning methodology that considers OoD images of individuals or small groups. To compute a metric-learning objective, we use the penultimate feature z of the in-distribution classifier $\mathcal{F}_{\text{in}}$ for an input x; we write z_{in} (resp. z_{ood}) as the feature of $x_{\text{in}} \in \mathcal{D}_{\text{in}}$ (resp. $x_{\text{ood}} \in \mathcal{D}_{\text{ood}}$). We train a classifier $\mathcal{F}$ to ensure that z_{in} is significantly different from z_{ood}, thereby preventing information overlap between the features. To realize this, a clustering-based metric learning objective is proposed.

Let $\mathcal{Z}_{\text{in}}$ and $\mathcal{Z}_{\text{ood}}$ be the sets of z_{in} and z_{ood}, respectively. We first construct a set of clusters $\mathcal{P}^{\text{in}}$ (resp. $\mathcal{P}^{\text{ood}}$) based on $\mathcal{Z}_{\text{in}}$ (resp. $\mathcal{Z}_{\text{ood}}$). Each cluster in $\mathcal{P}^{\text{in}}$ contains features of x_{in} corresponding to each class $c \in \mathcal{C}$, resulting in $|\mathcal{C}|$ clusters in $\mathcal{P}^{\text{in}}$. One straightforward way of constructing $\mathcal{P}^{\text{ood}}$ is to cluster images according to their incorrectly predicted classes. This, however, is sub-optimal in practice because such clusters are highly heterogeneous. For example, images of lakes and images of trees are semantically different, yet a cluster based on the "bird" class will contain both. Therefore, we construct $\mathcal{P}^{\text{ood}}$ by using a K-means clustering algorithm on $\mathcal{Z}_{\text{ood}}$.

We now have a set of clusters $\mathcal{P}^{\text{in}} = \{\mathcal{P}^{\text{in}}_c\}_{c=1}^{|\mathcal{C}|}$ and $\mathcal{P}^{\text{ood}} = \{\mathcal{P}^{\text{ood}}_k\}_{k=1}^{K}$. The center of each cluster is computed using $p_k = \frac{1}{|\mathcal{P}_k|}\sum_{x \in \mathcal{P}_k} z(x)$. We define the distance between the input image x and each cluster $\mathcal{P}_k$ as the distance between x's feature $z(x)$ and the center p_k, as follows:

$$d(x, \mathcal{P}_k) = \|z(x) - p_k\|_2 \quad (1 \le k \le K). \tag{4.1}$$

We design a loss $\mathcal{L}_{\text{d}}$ to ensure that the distance between x_{in} and in-distribution clusters $\mathcal{P}^{\text{in}}$ is small, but the distance between x_{in} and OoD clusters $\mathcal{P}^{\text{ood}}$ is large, as shown below:

$$\mathcal{L}_{\text{d}} = \sum_{c:y_c=1} d(x_{\text{in}}, \mathcal{P}^{\text{in}}_c) - \sum_{k \in \mathcal{K}} d(x_{\text{in}}, \mathcal{P}^{\text{ood}}_k), \tag{4.2}$$

where $y \in \{0,1\}^{|\mathcal{C}|}$ is the multi-hot binary vector of foreground classes in image x_{in} and $\mathcal{K}$ is the set of clusters in $\mathcal{P}^{\text{ood}}$ that are among the top-$\tau\%$ closest from x_{in}. This restriction of $\mathcal{K}$ ensures meaningful supervisory signals for the model.

We also use the usual classification loss $\mathcal{L}_{\text{cls}}$. For in-distribution samples x_{in}, we use the binary cross entropy (BCE) losses against the label vector y. For out-of-distribution samples x_{ood}, we use the same loss with the zero-vector label $y = (0, \cdots, 0)$. The classification loss for our classifier $\mathcal{F}$ is then

$$\mathcal{L}_{\text{cls}} = \frac{1}{|\mathcal{C}|}\sum_{c=1}^{|\mathcal{C}|}\left[\mathcal{L}_{\text{BCE}}(\mathcal{F}^c(x_{\text{in}}), y_c) + \mathcal{L}_{\text{BCE}}(\mathcal{F}^c(x_{\text{ood}}), 0)\right], \tag{4.3}$$

where $\mathcal{F}^c$ is the prediction for class c. The final loss $\mathcal{L}$ to train a classifier $\mathcal{F}$ is

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda\mathcal{L}_{\text{d}}, \tag{4.4}$$

where $\lambda > 0$ is a scalar balancing the two losses.

Because our method adds an additional regularization $\mathcal{L}_{\text{d}}$ to the existing classifier training, it can be seamlessly integrated into other methods, such as IRN [2], SEAM [107] and AdvCAM [112].

4.3.3 Training Segmentation Networks

The classifier $\mathcal{F}$ trained by Eq. 4.4 generates a localization map using the CAM [36] technique. Since the naive CAM generates low-resolution score maps and provides only rough localization of objects, recent WSSS methods [27, 107, 49, 157, 112, 48] have proposed a framework for expanding the CAM score map to full resolution. They consider the CAM localization map as an initial seed and generate pseudo-ground-truth masks by refining their initial seeds with established seed refinement methods [91, 29, 2, 119]. In this work, we apply the IRN framework [2] on our localization maps to obtain the pseudo-ground-truth masks. They are subsequently used for training segmentation networks.

Table 4.1: **W-OoD improves initial seeds.** We evaluate the qualities of various initial seeds and the effects of applygin W-OoD on them. Evaluated on Pascal VOC 2012 *train* set. All numbers are based on our re-implementation using the official codes.

Method	mIoU	Prec.	Recall	F1-score
IRN $_{\text{CVPR '19}}$ [29]	49.5	61.9	72.7	66.9
+ W-OoD	**53.3**	**66.5**	**73.2**	**69.7**
SEAM $_{\text{CVPR '20}}$ [107]	54.8	67.2	76.5	71.5
+ W-OoD	**55.9**	**68.5**	**76.7**	**72.4**
AdvCAM $_{\text{CVPR '21}}$ [112]	55.5	66.8	77.6	71.8
+ W-OoD	**59.1**	**71.5**	**77.9**	**74.6**

Table 4.2: **Quality of pseudo-GT masks.** Comparison of quality of the initial seed and pseudo ground-truth on the Pascal VOC 2012 *train* set. All the methods based based on IRN [2] with ResNet-50.

Method	Seed	+ CRF	Mask
IRN $_{\text{CVPR '19}}$ [2]	49.5	54.3	66.3
MBMNet $_{\text{ACMMM '20}}$ [131]	50.2	-	66.8
CONTA $_{\text{NeurIPS '20}}$ [48]	48.8	-	67.9
CDA $_{\text{ICCV '21}}$ [157]	50.8	-	67.7
AdvCAM $_{\text{CVPR '21}}$ [112]	55.6	62.1	69.9
CSE $_{\text{ICCV '21}}$ [167]	56.0	62.8	-
IRN + W-OoD (Ours)	53.3	58.4	71.1
AdvCAM + W-OoD (Ours)	**59.1**	**65.5**	**72.1**

4.4 Experiments

4.4.1 Experimental Setup

In-Distribution Dataset: We conduct experiments on the Pascal VOC 2012 [32] dataset. For the training images, we only use the image-level category labels, following the protocol for WSSS. We use the pixel-wise ground-truth masks on *val* (1,449 images) and *test* (1,456 images) sets only for evaluation. We use the official Pascal VOC evaluation server for the *test*-set evaluation.

Out-of-Distribution Dataset: As described in Sec. 4.3.1, we use the OpenImages [156] dataset to construct the candidate OoD set. As the result of prediction-score pruning and manual filtering, we obtain the hard OoD set $\mathcal{D}_{\text{ood}}$ with 5,190 images.

Reproducibility: We follow experimental settings of IRN [2] for training a classifier and obtaining the initial seed, including the use of ResNet-50 [135]. For the setting defined in Sec. 4.3.2, we use $\lambda = 0.007$, $\tau = 20$, and $K = 50$. For training a segmentation network, we use DeepLab-v2 [18] with two choices of backbones, ResNet-101 [135] and Wide ResNet-38 [136], following the practice in recent papers. All the backbones are pre-trained on ImageNet [3], following existing work [29, 107, 48, 167, 168].

4.4.2 Experimental Results

Quality of localization maps: As mentioned in Sec. 4.3.2, our method can be applied to other WSSS methods, since it only requires the addition of a loss term $\mathcal{L}_{\text{d}}$ during the classifier training. We apply our method to three state-of-the-art WSSS methods that utilize the initial seeds: IRN [2], SEAM [107], and AdvCAM [112]. Table 4.1 presents the qualities of the initial seeds for the considered baselines as well as respective performances when combined with our W-OoD technique. We observe that our method improves all the metrics by a large margin for all three methods. In particular, W-OoD training significantly improves precision values (+4.7%p for AdvCAM [112]),

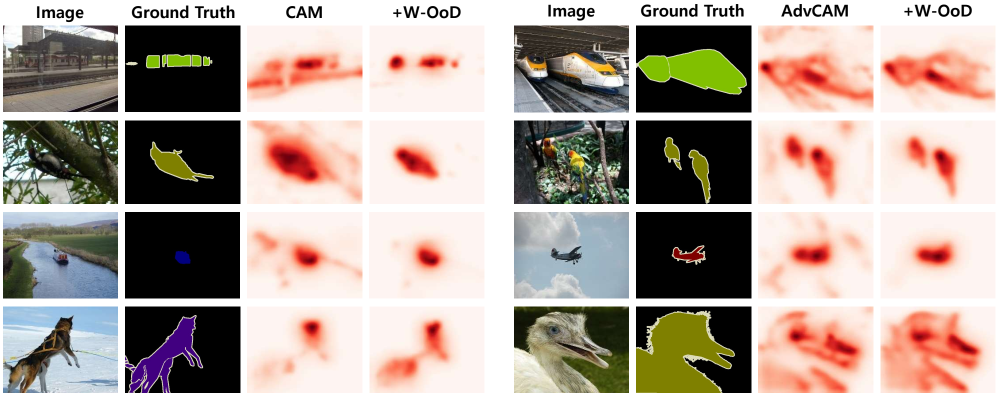


Figure 4.3: **Examples of localization maps.** The localization maps are obtained from CAM (left) and AdvCAM [112] (right). In each case, we show the results using our W-OoD method on top.

indicating that the resulting localization maps bleed into the background regions less frequently. This is what we expected to see as a result of including the hard OoD samples into training. Fig. 4.3 shows qualitative examples of the localization maps. They show that our method generates more precise maps around the actual foreground objects. Spuriously correlated background regions like rails for "train" and trees for "bird" are effectively suppressed by our method. Additionally, we observe that our method improves recall by expanding the retrieved region of the target object, as shown in the last column in Fig. 4.3. The increased precision gives room for further improvements in recall.

Quality of pseudo-ground-truth masks: Table 4.2 compares qualities of intermediate masks leading to the pseudo-ground-truth masks among state-of-the-art methods as well as ours. Our pseudo ground-truth masks achieve an mIoU value of 72.1, which outperforms the previous state of the art by a large margin. Note that CDA [157] is likewise motivated by the need to suppress spurious correlations between foreground and background cues, but has only used the in-distribution data to tackle the problem. It improves the initial seed of IRN [2] by 1.3%p mIoU (49.5 → 50.8), while our method

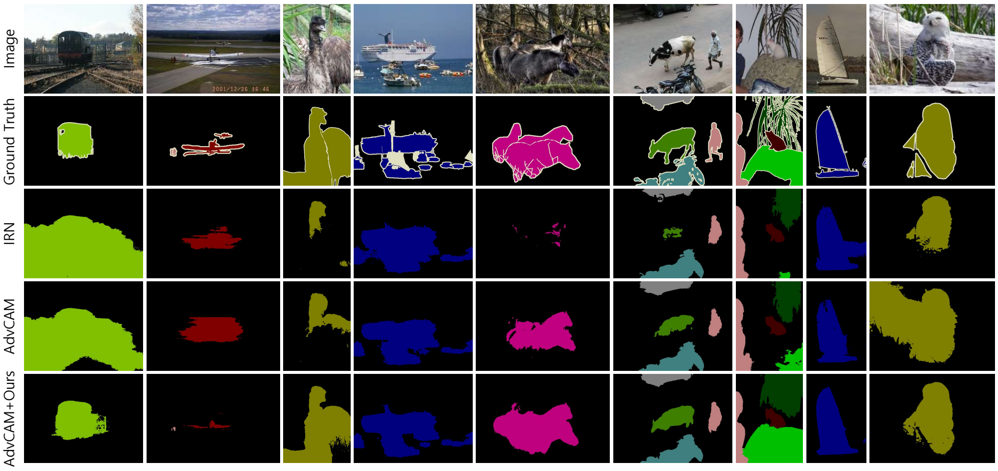


Figure 4.4: **Examples of final segmentation results.** Examples of semantic segmentation results on Pascal VOC 2012 *val* set for IRN [2], AdvCAM [112], and AdvCAM+Ours.

improves it by 3.8%p mIoU (49.5 → 53.3, in Table 4.1). We believe that in-distribution data are fundamentally limited in providing sufficient evidence for distinguishing certain background cues from foreground: if one always sees train on rail, how can one learn that rail is not part of the train? We believe this missing knowledge is effectively supplied by the hard OoD images.

Final segmentation results: We present the WSSS benchmark results in Table 5.3. It achieve the best result among the variants using only image-level tags: 70.7% mIoU on *val* and 70.1% mIoU on *test*. In particular, using the same backbone ResNet-101 [135], our method produces 2.3%p better mIoU than the baseline AdvCAM [112]. Our method also outperforms other methods using additional saliency supervision [92, 93] that explicitly provides pixel-level information of salient objects in an image, except for EDAM [163]. Fig. 4.4 shows examples of semantic masks produced by IRN [2], AdvCAM [112], and our AdvCAM + W-OoD. In the examples, our method captures the extent of the target objects more precisely than the baselines.

Table 4.3: **WSSS performance on Pascal.** We show results on Pascal VOC 2012 *val* and *test* sets. WResNet denotes Wide ResNet [136].

Method	Backbone	*val*	*test*
Supervision: Image-level tags + Saliency			
FickleNet $_{\text{CVPR '19}}$ [27]	ResNet-101	64.9	65.3
Sun *et al.* $_{\text{ECCV '20}}$ [94]	ResNet-101	66.2	66.9
A^2GNN $_{\text{TPAMI '21}}$ [169]	ResNet-101	68.3	68.7
AuxSegNet $_{\text{ICCV '21}}$ [170]	WResNet-38	69.0	68.6
EDAM $_{\text{CVPR '21}}$ [163]	ResNet-101	70.9	70.6
Supervision: Image-level tags			
IRN $_{\text{CVPR '19}}$ [2]	ResNet-50	63.5	64.8
SEAM $_{\text{CVPR '20}}$ [107]	WResNet-38	64.5	65.7
CONTA $_{\text{NeurIPS '20}}$ [48]	WResNet-38	66.1	66.7
AdvCAM $_{\text{CVPR '21}}$ [112]	ResNet-101	67.5	67.1
CSE $_{\text{ICCV '21}}$ [167]	WResNet-38	68.3	68.0
PMM $_{\text{ICCV '21}}$ [168]	WResNet-38	68.5	69.0
AdvCAM + W-OoD (Ours)	ResNet-101	69.8	69.9
AdvCAM + W-OoD (Ours)	WResNet-38	**70.7**	**70.1**

4.4.3 Analysis and Discussion

Number of OoD Images We investigate the impact of the number of OoD images for our W-OoD training method. Fig. 4.5(a) shows the mIoU scores of the initial seed at different numbers of OoD images ($|\mathcal{D}_{\text{ood}}|$) while keeping the number of in-distribution images constant at $|\mathcal{D}_{\text{in}}| = 10,582$. The experiments were repeated five times to investigate the sensitivity of the result to different random subsets of $\mathcal{D}_{\text{ood}}$. We observe that already at 1 hard OoD sample per class ($|\mathcal{D}_{\text{ood}}| = 20$), the performance boost is 2.0%p (49.8 $\rightarrow$ 51.8), though with a significant amount of variance. The marginal gain

K	Clustering	mIoU
20	Predicted classes	52.1
20	K-Means	52.4
30		53.1
50		**53.3**
70		52.6

Table 4.4: **Constructing $\mathcal{P}^{\text{ood}}$.** We compare two methods for constructing $\mathcal{P}^{\text{ood}}$ for W-OoD training. We report the mIoU of the initial seeds on Pascal VOC 2012 *train* set.

from additional hard OoD images diminishes with increasing number of samples. The performance variance also diminishes with an increased number of hard OoD samples.

In the second experiment, we vary the number of hard OoD samples $|\mathcal{D}_{\text{ood}}|$ while fixing the total number of image-level labeled samples: $|\mathcal{D}_{\text{in}}| + |\mathcal{D}_{\text{ood}}| = 10,582$. This is a version of fixing the budget for in-distribution and out-of-distribution samples. Fig. 4.5(b) shows that the hard OoD images bring far greater unit gain than in-distribution images. Thus, given a fixed budget, it is advisable to spend at least some portion of it on collecting the hard OoD samples.

In Fig. 4.5(c), we observe that, with 100 hard OoD images, we only need 2,000 in-distribution images to match the performance we obtain from the original 10,582 in-distribution images, enhancing the data efficiency by around 500$\%$.

Effectiveness of K-Means clustering: Table 4.4 compares the two methods for constructing the $\mathcal{P}^{\text{ood}}$ in Sec. 4.3.2. When the OoD clusters are based on the classes predicted by the classifier, the resulting mIoU is 52.1$\%$, which is not significantly different from that obtained using the K-means clustering method for the same K value. The clustering method based on the predicted class limits K to $|\mathcal{C}|$, whereas K values can be controlled in K-means clustering. At $K = 50$, it produces an mIoU value of 53.3$\%$ and the performance is stable across a broad range of K values. Examples of

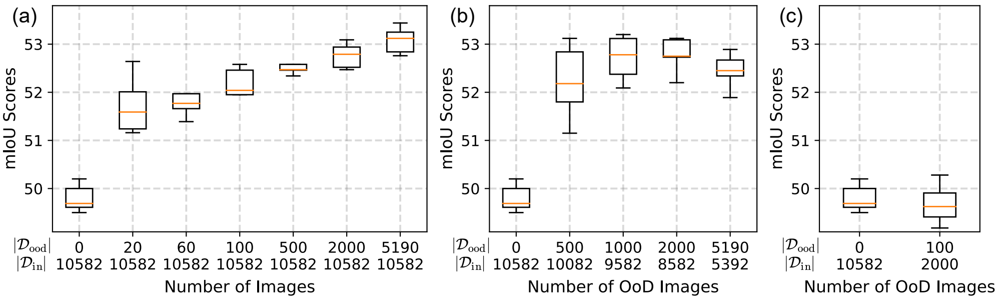


Figure 4.5: **Amount of hard OoD samples.** We vary number of in-distribution training data $\mathcal{D}_{\text{in}}$ (originally 10,582) and the hard OoD data (originally 0). (a) We fix $|\mathcal{D}_{\text{in}}| = 10,582$ and vary $|\mathcal{D}_{\text{ood}}|$. (b) We fix $|\mathcal{D}_{\text{in}}| + |\mathcal{D}_{\text{ood}}| = 10,582$ and vary $|\mathcal{D}_{\text{ood}}|$. (c) We use $|\mathcal{D}_{\text{in}}| = 2,000$ and $|\mathcal{D}_{\text{ood}}| = 100$. The box plots show the quantiles over five repeated experiments.

OoD samples in each cluster are presented in the Appendix.

Effectiveness of Each Loss function: We conduct ablation studies for each loss in Eq. 4.4. Both $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{d}}$ consist of terms for in-distribution $\mathcal{D}_{\text{in}}$ and out-of-distribution $\mathcal{D}_{\text{ood}}$ data. The effectiveness of each loss term as well as the dataset type is presented in Table 4.5. (a) is the result of using only $\mathcal{L}_{\text{cls}}$ for $\mathcal{D}_{\text{in}}$, which is our baseline. The performance boost for (a)→(b) and (c)→(e) indicates that training the classifier to predict OoD images as background ($\mathcal{L}_{\text{cls}}$ on $\mathcal{D}_{\text{ood}}$) is effective, though with only marginal improvements. The improvement along (b)→(d)→(f) signifies the importance of $\mathcal{L}_{\text{d}}$, in particular when used on the hard OoD data $\mathcal{D}_{\text{ood}}$. We also find that $\mathcal{L}_{\text{d}}$ for $\mathcal{D}_{\text{in}}$ is useful for stabilizing the performance: in (e)→(f), the standard deviation decreases from 0.82 to 0.33.

Analysis of Results by Class Different object classes exhibit different amounts of spurious correlation with background. For example, "train" objects are often confused with the rail background due to their high co-occurrence with rails. Objects like "tv-monitor", on the other hand, suffer less from this issue because of the variety of the co-occuring concepts: a TV can be freely put next to a wall, furniture, window, or any

Table 4.5: **Loss ablations.** Effectiveness of each loss on the initial seed in mIoU(%) on Pascal VOC *train* set.

Loss	Data	(a)	(b)	(c)	(d)	(e)	(f)
$\mathcal{L}_{\text{cls}}$	$\mathcal{D}_{\text{in}}$	✓	✓	✓	✓	✓	✓
	$\mathcal{D}_{\text{ood}}$		✓		✓	✓	✓
$\mathcal{L}_{\text{d}}$	$\mathcal{D}_{\text{in}}$			✓	✓		✓
	$\mathcal{D}_{\text{ood}}$			✓		✓	✓
mIoU		49.5	50.0	52.5	50.2	52.3	53.3

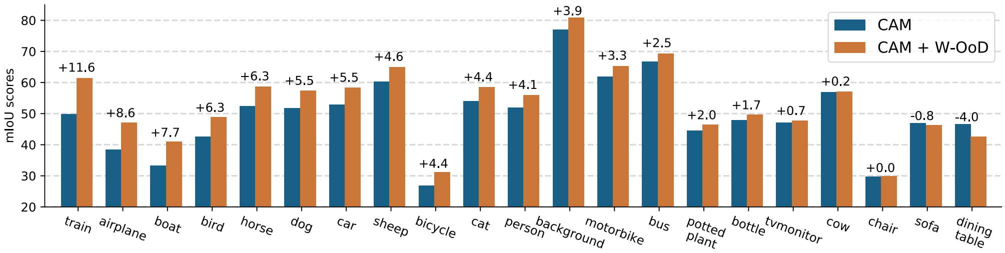


Figure 4.6: **Per-class seed qualities.** We compare the baseline IRN [2] (denoted as "CAM" above) and the W-OoD augmented version for each class. Evaluated on Pascal VOC 2012 *train* set. Classes are sorted in the descending order of ∆improvement (%p).

other indoor objects. We show the class-wise performances for the baseline IRN [2] and ours in Fig. 4.6. First of all, we note that our method improves the class-wise performances rather proportionately: 18 out of 21 classes have seen a performance improvement. Classes that have benefited most from our method are train, airplane, boat, bird, and horse. They are ones that are well-known for spurious background correlations: train-rail, airplane-sky/runway, boat-water, bird-tree/sky, and horse-meadow.

On the other hand, a particularly large drop in mIoU is seen for the "dining table" class. We conjecture the spurious background correlation has actually been helping out the localization of the "dining table" objects. Many pixel-wise ground-truth evaluation mask for "dining table" objects erroneously include the objects put on it, such as

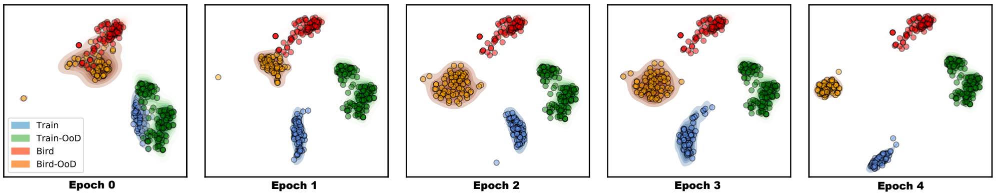


Figure 4.7: **Visualization of intermediate features.** We visualize the intermediate features for "train" and "bird" classes, as well as the features for respective OoD samples, at different training stages. We use the T-SNE [141] dimensionality reduction technique.

plates, cutlery, and foods. By labeling OoD images, which contain those co-occurring objects not put on a dining table, as "no dining table", the model may correctly assign lower "dining table" scores on those objects, ironically harming the final performance measured on noisy masks. See Appendix for the examples. We believe there will be an additional performance gain if those wrong ground-truth masks are fixed.

Manifold Visualization To observe the training dynamics of our method, we visualize the feature manifold at different stages of the W-OoD training. We collect two sets of images with respective labels "train" and "bird" from $\mathcal{D}_{\text{in}}$ and two sets of images which are respectively falsely predicted as "train" and "bird" by $\mathcal{F}_{\text{in}}$ from $\mathcal{D}_{\text{ood}}$. Using the classifier at epoch $e \in \{0, \cdots 5\}$[1], we compute the features z_{in} and z_{out} from images drawn from $\mathcal{D}_{\text{in}}$ and $\mathcal{D}_{\text{ood}}$, respectively. We use t-SNE [141] to reduce the dimensionality of each feature to 2 dimensions. Fig. 4.7 visualizes the features z_{in} and z_{out} after dimensional reduction using t-SNE. It is observed that, at the beginning of the epoch, z_{in} and z_{out} of each class are rarely distinguishable, indicating that the classifier encodes similar information for in-distribution and OoD images. However, as W-OoD training progresses, the two features gradually become distinct. This analysis supports the argument that our method allows the classifier to avoid modeling common information between in-distribution and OoD images, as intended.

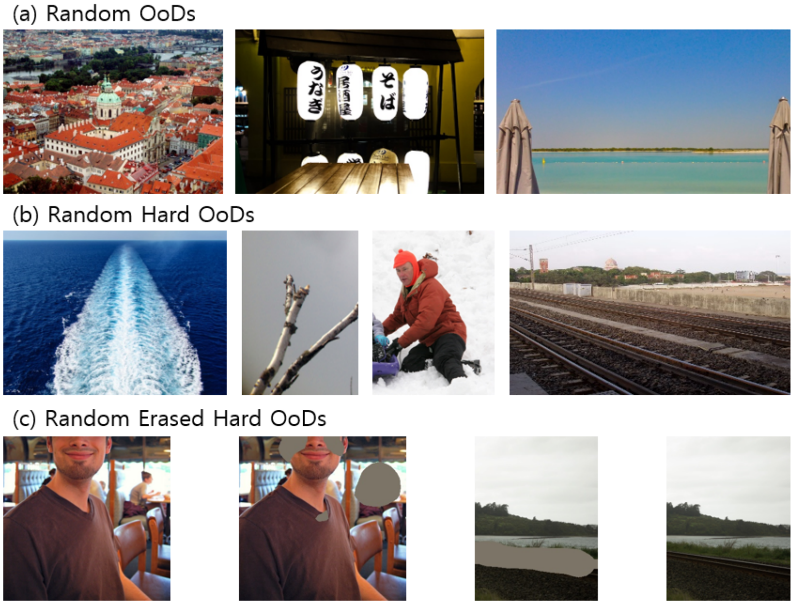
[1] The classifier at $e = 0$ is the one trained using in-distribution images.

OoD Collection Method	mIoU
No OoDs	49.5
Random OoDs	51.9
Random Hard OoDs	51.1
Random Hard OoDs + Erasure	51.4
Manual Filtering	53.3

Table 4.6: Comparison of alternative OoD collecting methods.

4.5 Analysis of OoD Collection Process

Manual filtering introduced in Section 4.3.1 is essential in the OoD collection process to filter out the samples that are not actually OoDs, which is caused by the label noise issue in OpenImages [156] dataset. Manual filtering is efficient because it requires just yes/no answer whether the given candidate OoD image contains one of the foreground classes or not, but it still requires additional human labour. Therefore, we test some alternative approaches to construct the OoD dataset and report their results in Table 4.6. For a fair comparison, we fix the total number of images in the dataset for all approaches. First, we try to use random OoDs, which are directly obtained from the candidate OoDs. It produces 2.4%p mIoU gain over the baseline, but cannot reach the performance of manual filtering. We then use the random hard OoDs, which are the samples confused by a classifier to be containing the foreground object. This set contains more label noise than random OoDs, because these samples are likely to contain missing positives, *i.e.,* in-distribution images. Therefore, it produces inferior result compared to the random OoDs. To remove the foreground information in hard OoDs, we finally try to remove class-relevant regions in random hard OoDs to guarantee that the collected OoDs do not contain foreground information. We compute the CAM for hard OoDs and remove the corresponding regions similar to the erasure methods [86, 90]. Examples of OoDs for each alternative method are shown in Figure 4.8.



(a) Random OoDs

(b) Random Hard OoDs

(c) Random Erased Hard OoDs

Figure 4.8: Examples of (a) Random OoDs, (b) Random Hard OoDs, and (c) Random Erased Hard OoDs.

The erasure approach produces better performance than random hard OoDs, because has higher precision by removing class-relevant regions, but lower performance than random OoDs. We present three possible explanations for this phenomenon. First, if the class-relevant regions are erased in hard OoD, meaningful information that can distinguish foreground and background is deleted, so it is no longer hard ood. Second, it is not trivial to determine the threshold of CAM. Third, forcibly erasing regions of the image makes metric learning trivial because the distribution of the image changes.

4.6 Integrating Proposed Methods

We have proposed three methods based on class labels in Sections 3 and 4: W-OoD [171], AdvCAM [112], and RIB [49]. We now analyze whether these methods can have a complementary relationship with each other. Table 4.7 presents the experimental results obtained by combining these methods. We can see that W-OoD can be successfully

Method	mIoU
Baseline	49.5
+W-OoD (§4)	53.3
+W-OoD (§4) + AdvCAM (§3.3)	59.1
+W-OoD (§4) + RIB (§3.4)	56.8
+W-OoD (§4) + AdvCAM (§3.3) + RIB (§3.4)	60.7

Table 4.7: Performance obtained by successively integrating the proposed method.

integrated with either AdvCAM or RIB, producing significant improvements (AdvCAM brings 5.8%p gain and RIB brings 3.5%p gain). However, we observe that integrating all the methods brings just 1.6%p over W-OoD + AdvCAM. Since AdvCAM and RIB share a similar goal of increasing the value of class logit, the performance gain from using them together is less than that of adding them individually. However, they can still work in tandem, resulting in a 1.6% mIoU gain.

4.7 Summary

We have proposed the use of a new source of information, the OoD data, for suppressing the spurious correlations learned by weakly supervised semantic segmentation (WSSS) methods. We have showcased the data collection pipeline whereby the suitable hard OoD images are obtained. By including those images as negative samples in addition to the original in-distribution foreground samples, we have been able to train a classifier with more accurate localization maps. Our method achieves a performance superior to existing WSSS methods based on image-level labels. In addition, we have empirically shown that the image-level labeling cost itself can be further reduced by using the hard OoD images, without sacrificing the WSSS performances. We have focused on using OoD images for training classifiers to produce accurate pseudo ground-truth masks; interesting future work will include exploiting the OoD images in training a segmentation network itself.

Chapter 5

Learning with Bounding Box Labels

5.1 Introduction

Object segmentation is one of the most important steps in image recognition. Advances in deep learning have greatly improved the performance of semantic and instance segmentation [20, 18] through the use of huge amounts of pixel-level annotated training data. However, annotating with pixel-level masks requires a lot of effort. According to Bearman *et al.* [21], constructing a pixel-level mask for an image containing an average of 2.8 objects takes about 4 minutes. This is why weakly supervised methods have been proposed, in which segmentation networks are trained using annotations that are less detailed than pixel-level masks, such as bounding boxes [25, 24, 98], or image-level tags [27, 2, 29].

The most easily obtainable annotation is the class label. Labeling an image with class labels takes around 20 seconds [21], but it only indicates that objects of certain classes are depicted and gives no information about their locations in the image. Moreover, class labels provide no help in separating different objects of the same class, which is the goal of instance segmentation.

Bounding boxes provide information about individual objects and their locations. Bounding box annotation takes about 38.1 seconds per image [55], which is much more attractive than constructing pixel-level masks. Many researchers have tackled semantic segmentation [98, 24, 25, 99] and instance segmentation [24, 100, 101, 102, 103] using bounding box annotations as a search space in which a class-agnostic object mask can be found by an off-the-shelf object mask generator. These are mostly based on GrabCut [56] or multiscale combinatorial grouping (MCG) [57]. Those mask generators operate on the low-level information of images, such as the color or brightness of pixels, and this limits the quality of the resulting mask. Thus, applying these mask generators to bounding box annotations requires additional steps such as estimating what proportion of the pixels in a bounding-box belong to the corresponding object [25, 99], iterative refinement of an estimated mask [98], and auxiliary attention modules [99].

We propose a pixel-level method of localizing a target object inside its bounding box using a trained object detector. We make use of attribution maps obtained from the trained object detector, which highlight the image regions that the detector focuses on in conducting object detection. Inspired by the perturbation methods used to explain the output of image classifiers [172, 173, 174], we introduce a bounding box attribution map (BBAM) which provides an indication of the smallest areas of an image that are sufficient to make an object detector produce almost the same result as that from the original image. The BBAM identifies the area occupied by the object in each bounding box predicted by the trained object detector. Since this localization takes place at the pixel level, it can be used as a pseudo ground truth for weakly supervised learning of semantic and instance segmentation.

The main contributions of this chapter can be summarized as follows.

- We propose a bounding box attribution map (BBAM), which can draw on the rich semantics learned by an object detector to produce pseudo ground-truth for training semantic and instance segmentation networks.
- Our technique significantly outperforms previous state-of-the-art methods of weakly supervised semantic and instance segmentation, assessed on the PASCAL VOC 2012 and MS COCO 2017 benchmarks.

- We analyze our method from various viewpoints, providing deeper insights into the properties of the BBAM.

5.2 Related Work

Fully supervised semantic and instance segmentation based on pixel-level annotations is highly reliable, but the manual annotation process is laborious. This requirement is overcome by weakly supervised methods based on inexact, but easily obtainable, annotations such as scribbles [23], bounding boxes [25, 24], or class labels [27, 2, 94]. In this section, we briefly review some recently introduced weakly supervised approaches that use class labels or bounding boxes. In addition, we describe some visual saliency methods related to our method.

Learning with Class Labels A class activation map (CAM) [36] is a widely adopted technique to obtain a localization map from class labels. However, a CAM only identifies the most discriminative regions of objects [27, 33], and hence the majority of existing methods that use class labels [33, 27, 47, 91, 29, 90, 59, 28, 175, 118, 34] are primarily concerned with expanding the area of the target object activated by a CAM. For instance, erasure methods [90, 86] iteratively find new regions of the target object by removing discriminative regions in an image. Other methods [118, 94] consider the information shared between several images by capturing cross-image semantic similarities and differences. Seed growing and refinement techniques [29, 2, 91] are typically used to expand the regions representing the target object imperfectly that are in the initial CAM, on the basis of relationships between pixels. Other methods construct CAMs that embody the multi-scale semantic context in an image [27, 59, 108]. Despite these efforts, the information available from class labels remains limited, so auxiliary information acquired from web images [35] or videos [34, 33] can be used together.

Learning with Bounding Boxes Class labels have led to significant achievements in semantic segmentation, but they are inherently unhelpful in instance segmentation, which requires the separation of different objects of the same class. In contrast, bounding boxes do provide information about the location of individual objects in an image, and they are still much cheaper than constructing pixel-level masks [55]. Most existing methods utilized a bounding box as a search space to conduct low-level searches for object masks. They create a pseudo mask within a box using off-the-shelf methods of mask proposal such as MCG [57] or GrabCut [56]. These processes can be guided by specifying the proportion of the pixels in a bounding box that are likely to belong to the object [25, 99]. Iterative mask refinement techniques [98] can also be applied. However, these methods are largely based on low-level information in the image, and they ignore the semantics associated with the bounding boxes. A rare exception is the multiple-instance learning formulation with a bounding box tightness prior [102]: a crossing line within a box must contain at least one pixel of the target object. The drawback with this approach is that only a small number of pixels are contributing to the localization of the object.

Visual Saliency Methods Various methods have been proposed to visually explain the predictions of deep neural networks (DNNs) [172, 173, 176, 177, 36] in a form of a saliency map. However, most studies have been concerned with classifiers, and only a few have looked at DNNs performing other tasks [178, 179]. In particular, there have been no attempts to explain the predictions of object detectors, except Wu *et al.* [180], who embedded interpretability inside the DNN, in this case Faster R-CNN [15]. However, the explanation produced by their modified DNN is not immediately understandable because it is given as a form of tree, and thus it is not appropriate to generate pseudo ground truth for weakly supervised segmentation. Gradient-based methods, such as SimpleGrad [181], SmoothGrad [182], and Grad-CAM [37], can provide visual saliency maps of the results from classifiers, but these methods are

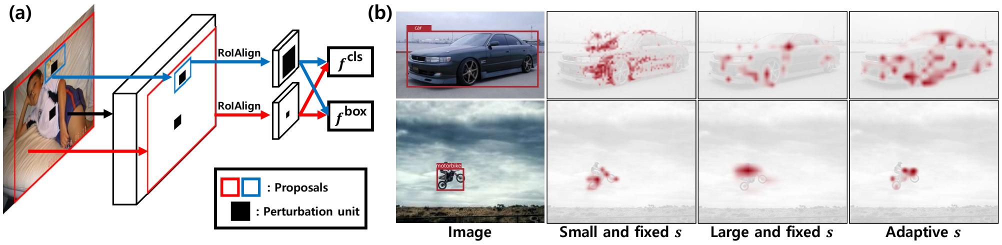


Figure 5.1: The size of the *perturbation unit* needs to be adjusted to the object size. (a) RoIAlign [20] produces perturbation units of different sizes. (b) Examples of resulting BBAMs with small fixed values of s, large fixed values of s, and values of s determined adaptively. Fixed values of s, whether large or small, tend to generate unwanted artifacts.

not easily extended to object detectors, because of the structural difference between classifiers and object detectors. Nevertheless, gradient-based methods have a significant bearing on our approach, and we look at them in more detail in Section 5.5.

5.3 Methods

5.3.1 Revisiting Object Detectors

Modern object detectors can be fallen into two categories: one-stage [183, 65, 64] and two-stage [15, 63] approaches. We focus on two-stage object detectors such as Faster R-CNN [15], in which the two stages are region proposal and box refinement. A region proposal network (RPN) generates candidate object proposals in the form of bounding boxes; but these proposals are class-agnostic and noisy, and most of them are redundant, thereby necessitating a subsequent refinement step, in which classification and bounding box regression are performed on each proposal. Since the proposal boxes proposed by the RPN are of different sizes, RoI pooling (*e.g.,* RoIAlign [20]) is used to convert the feature map corresponding to each proposal to a predefined fixed size, as shown in Figure 5.1(a). The pooled feature map is then passed to the *classification head* and also to the *bounding box regression head*.

Classification head. It computes the class probability p^c of class c for each proposal and assigns the most likely class $c^* = \mathrm{argmax}_c\, p^c$ to the proposal.

Bounding box regression head. It adjusts the noisy proposal to fit the object by computing the offsets $t^c = (t_x^c, t_y^c, t_w^c, t_h^c)$ for each class $c \in \{1, 2, \cdots, C\}$. The final localization is obtained by shifting each coordinate of the proposal using the offset t^{c^*}. We refer to Ren *et al.* [15] for the details of the parameterization of each coordinate.

For simplicity, we will abbreviate *classification head* and *bounding box regression head* as *cls head* and *box head*, respectively.

5.3.2 Bounding Box Attribution Map

Suppose we are given an image I and the corresponding bounding box annotations. We also have a set of object proposals $\mathcal{O} = \{o_k\}_{k=1}^K$, either given or obtained by RPN, where K is the number of proposals. For each proposal o_k, the *box head* f^{box} and the *cls head* f^{cls} produce box offsets $t_k = f^{\text{box}}(I, o_k)$ and the class probability $p_k = f^{\text{cls}}(I, o_k)$, respectively. We omit the proposal indices k for brevity.

The bounding box attribution map (BBAM) identifies the important region in the image that the detector needs to perform object detection. We find the smallest mask $\mathcal{M} : \Omega \rightarrow [0, 1]$ where Ω is a set of pixels, which captures a subset of the image that produces almost the same prediction as the original image. A small $\mathcal{M}$ reduces the amount of unnecessary information reaching the detector. The mask specifies a subset of the image in terms of the perturbation function $\Phi(I, \mathcal{M}) = I \circ \mathcal{M} + \mu \circ (1 - \mathcal{M})$, where $\circ$ denotes pixel-wise multiplication, and μ is the per-channel mean of the training data with the same size as $\mathcal{M}$. For each proposal o, the best mask $\mathcal{M}^*$ is obtained by optimizing the following function using gradient descent with respect to $\mathcal{M}$:

$$\mathcal{M}^* = \underset{\mathcal{M} \in [0,1]^\Omega}{\mathrm{argmin}}\ \lambda \left\|\mathcal{M}\right\|_1 + \mathcal{L}_{\text{perturb}}, \tag{5.1}$$

$$\begin{aligned} \mathcal{L}_{\text{perturb}} = \ & \mathbb{1}_{\text{box}} \left\| t^c - f^{\text{box}}(\Phi(I, \mathcal{M}), o) \right\|_1 \\ & + \mathbb{1}_{\text{cls}} \left\| p^c - f^{\text{cls}}(\Phi(I, \mathcal{M}), o) \right\|_1, \end{aligned} \tag{5.2}$$

where $\mathbb{1}_{\text{box}}$ and $\mathbb{1}_{\text{cls}}$ are logical variables that have a value of 0 or 1, to control which head is used to produce localizations, and $t^c = f^{\text{box}}(I, o)$ and $p^c = f^{\text{cls}}(I, o)$ are the predictions for the original image.

Previous studies show that using a mask of the same spatial size as the input image incurs undesirable artifacts due to the adversarial effect [38]: even a perturbation in a tiny magnitude can significantly change the prediction of a DNN. This problem can be addressed by introducing a coarse mask downsampled by a stride s [172, 173, 174, 178], so multiple image pixels are perturbed by a single element of $\mathcal{M}$. We can then optimize $\mathcal{M} \in \mathbb{R}^{\lceil w/s \rceil \times \lceil h/s \rceil}$ for the image $I \in \mathbb{R}^{w \times h}$, using the perturbation function $\Phi(I, \mathcal{M}) = I \circ \hat{\mathcal{M}} + \mu \circ (1 - \hat{\mathcal{M}})$, where $\hat{\mathcal{M}} \in \mathbb{R}^{w \times h}$ is upsampled $\mathcal{M}$ to a width of w pixels and a height of h pixels.

Existing methods of explaining the output of classifiers [172, 173, 174] or semantic segmentation networks [178] use a fixed value of s for all images, *i.e.*, they fix the size of a *perturbation unit*[1]. However, in the case of object detectors, a *perturbation unit* of fixed size can result in perturbations of different sizes to the RoI-pooled features, depending on the size of the proposals, as shown in Figure 5.1(a). Figure 5.1(b) shows how the size of a *perturbation unit*, after RoI pooling, can fail to match the sizes of target objects: the perturbations are too coarse for small objects and too fine for large objects. Therefore, we use an adaptive stride $s(a)$ where a is the ratio of the area of the bounding box predicted by the object detector to that of the image, so that we use a small stride for a small object and a large stride for a large object.

5.3.3 Training the Segmentation Network

Generating Pseudo Ground Truth Since the BBAM is a pixel-level localization of the target object in a bounding box predicted by the object detector, it can be used as pseudo ground-truth for weakly supervised semantic and instance segmentation, using the following procedure: We first train an object detector, then create pseudo

[1] The *perturbation unit* is a block of image pixels perturbed by a single element of $\mathcal{M}$.

ground-truth semantic and instance masks for training images, using the BBAM of the trained object detector. These pseudo ground-truth masks can then be used to train semantic and instance segmentation networks. We will now explain this procedure in more detail.

Creating masks. Multiple proposals on a single object yield multiple predictions from the object detector. In order to benefit from the diversity of these predictions, we build the pseudo ground-truth from the BBAMs of multiple proposals. For each ground-truth box, we generate a set of object proposals $\mathcal{O}$ by randomly jittering each coordinate of the box by up to $\pm 30\%$. These proposals are sent to the f^{cls} and the f^{box}. If the f^{cls} correctly predicts the ground-truth class, and the intersection over union (IoU) value associated with the predicted box by f^{box} is greater than 0.8, then the proposal is added to a set of positive proposals $\mathcal{O}^{+} \subset \mathcal{O}$. We then use a modified version of $\mathcal{L}_{\text{perturb}}$ in Eq. 5.1 to amalgamate all the positive proposals into a single localization map, as follows:

$$\begin{aligned}\mathcal{L}_{\text{perturb}} = \mathbb{E}_{o \in \mathcal{O}^{+}}[\mathbb{1}_{\text{box}} \left\| t^{c} - f^{\text{box}}(\Phi(I, \mathcal{M}), o) \right\|_{1} \\ + \mathbb{1}_{\text{cls}} \left\| p^{c} - f^{\text{cls}}(\Phi(I, \mathcal{M}), o) \right\|_{1}].\end{aligned} \tag{5.3}$$

In this equation both $\mathbb{1}_{\text{box}}$ and $\mathbb{1}_{\text{cls}}$ are set to 1, since the BBAMs of f^{box} and f^{cls} provide complementary localization results (see Section 5.5 for details). A BBAM obtained in this way may partially cover the target object because not all pixels of the object are considered by f^{box} and f^{cls}. Therefore we refine the BBAM using CRFs [75], following previous work [25, 29, 24]. Finally, we create pseudo instance-level ground-truth masks by considering the pixels in each BBAM with values greater than a threshold θ to be foreground. We denote such a mask as $\mathcal{T}$.

The threshold θ controls the size of $\mathcal{T}$. However, the proportion of pixels in each BBAM which correspond to the foreground will vary, so it may not be appropriate to use a fixed θ. Therefore we introduce two thresholds θ_{fg} and θ_{bg}: pixels whose attribution values are higher than θ_{fg} are considered to be part of the foreground, and pixels whose values are lower than θ_{bg} are considered to be part of the background. The remaining pixels are ignored in the loss computations during training segmentation networks.

Refine with MCG proposals. MCG [57] is an unsupervised mask proposal generator, which is commonly used in weakly supervised instance segmentation [184, 185, 103, 95, 24]. We can use mask proposals generated by MCG to refine a mask $\mathcal{T}$. We first select the mask proposal that has the highest IoU with $\mathcal{T}$. However, that proposal may partially cover the target object. We therefore consider other proposals that are completely contained within $\mathcal{T}$. More formally, given a set of MCG proposals $\{m_i\}_{i=1}^{K}$, the refined mask $\mathcal{T}_r$ is derived as follows:

$$\begin{aligned} &\mathcal{T}_r = \bigcup_{i \in \mathcal{S}} m_i, \quad \text{where} \\ &\mathcal{S} = \{i \,|m_i \subset \mathcal{T}\} \cup \{\underset{i}{\operatorname{argmax}} \operatorname{IoU}(m_i, \mathcal{T})\}. \end{aligned} \tag{5.4}$$

We now explain the procedure that we use for training the semantic and instance segmentation network.

Instance segmentation. We use Mask R-CNN [20], pre-trained on ImageNet [3]. We use a seed growing technique [91, 29, 27, 33] for pseudo-labeling the pixels ignored during training: Starting with the pixels identified by the initial pseudo ground-truth mask, more of the ignored pixels progressively participate in the loss computation as training proceeds. We refer to Huang *et al.* [91] for more details.

Semantic segmentation. We use DeepLab-v2 [18], pre-trained on the ImageNet [3] dataset. The pseudo labels produced in the previous section can easily be made suitable for semantic segmentation by converting them from instance-level to class-level. Pixels assigned to two or more object classes are ignored during the loss computation.

5.4 Experiments

5.4.1 Experimental Setup

Dataset and evaluation metrics. We conducted experiments on the PASCAL VOC [32] and the MS COCO datasets [79]. The PASCAL VOC dataset contains 20 object classes and one background class. Following the same protocol as other recent work on weakly supervised semantic and instance segmentation [2, 103, 102, 25], we used an augmented set of 10,582 training images produced by Hariharan *et al.* [133]. The MS COCO dataset has 118K training images containing 80 object classes. We report mIoU values for semantic segmentation. For instance segmentation, we report average precision (AP$_\tau$) at IoU thresholds τ; averaged AP over IoU thresholds from 0.5 to 0.95; and the average best overlap (ABO).

Reproducibility. We used the PyTorch [116] implementation [186] of Faster R-CNN [15] and Mask R-CNN [20]. For semantic segmentation, we used the PyTorch implementation of DeepLab-v2-ResNet101 [150]. We set $s(a)$ to $16 + 48\sqrt{a}$ and λ to 0.007. We set θ_{fg} and θ_{bg} to 0.8 and 0.2 respectively. To find $\mathcal{M}^*$ in Eq. 5.1, we used Adam optimizer [187] with a learning rate of 0.02 for 300 iterations. The experiments were performed on NVIDIA Tesla V100 GPUs. For MCG mask proposals, we used the pre-computed proposals for PASCAL VOC and MS COCO images provided by Pont-Tuset *et al.* [57].

5.4.2 Weakly Supervised Instance Segmentation

Results on PASCAL VOC. Table 5.1 compares the performance of weakly supervised instance segmentation by using image-level tags or bounding boxes. Our method significantly outperforms those methods. Specifically, the AP$_{50}$ and AP$_{70}$ values of our method are both 6.0% higher than those of the previous best performing method which also uses bounding box annotation [103]. We include results from two fully supervised methods: MNC [188] and Mask R-CNN [20]. The performance of Mask R-CNN [20], which is fully supervised, can be viewed as an upper bound on the achievable performance of our method. We achieve 92.2% and 95.7% of the performance of fully supervised Mask R-CNN, in terms of AP$_{50}$ and ABO respectively. Figure 5.2 presents examples of instance masks produced by our method.

Results on MS COCO 2017. This is a challenging dataset containing more objects

Table 5.1: Weakly supervised instance segmentation performance on PASCAL VOC 2012 *val* images.

Method	AP_{25}	AP_{50}	AP_{70}	AP_{75}	ABO
Full supervision: Instance masks					
MNC $_{\text{CVPR '16}}$ [188]	-	63.5	41.5	-	-
Mask R-CNN $_{\text{ICCV '17}}$ [20]	77.3	69.1	49.9	41.9	65.8
Weak supervision: Image-level tags					
PRM $_{\text{CVPR '18}}$ [184]	44.3	26.8	-	9.0	37.6
IRNet $_{\text{CVPR '19}}$ [2]	-	46.7	23.5	-	-
LIID $_{\text{TPAMI '20}}$ [95]	-	48.4	-	24.9	50.8
Arun *et al.* $_{\text{ECCV '20}}$ [103]	59.1	49.7	29.2	27.1	-
Weak supervision: Bounding boxes					
SDI $_{\text{CVPR '17}}$ [24]	-	44.8	-	16.3	49.1
Liao *et al.* $_{\text{ICASSP '19}}$ [100]	-	51.3	-	22.4	51.9
Sun *et al.* $_{\text{Access '20}}$ [101]	-	56.9	-	21.4	56.9
Hsu *et al.* $_{\text{NeurIPS '19}}$ [102]	75.0	58.9	30.4	21.6	-
Arun *et al.* $_{\text{ECCV '20}}$ [103]	73.1	57.7	33.5	31.2	-
BBAM (Ours)	**76.8**	**63.7**	**39.5**	**31.8**	**63.0**

in an image on average than PASCAL VOC. The sizes of instances of objects are also more diverse. Table 5.2 compares the performance of our method with that of other weakly supervised instance segmentation methods with various levels of supervision on MS COCO. Our method achieves a 6.7% higher value of AP_{75} than the previous best performing method which uses bounding box annotations. Since the labels for *test-dev* images are not publicly available, the results for the *test-dev* images were obtained from the MS COCO challenge website.

Table 5.2: Comparison of instance segmentation methods with various types of supervision on MS COCO.

Method	sup.	AP	AP$_{50}$	AP$_{75}$
MS COCO *val* images				
Mask R-CNN $_{\text{ICCV '17}}$ [20]	$\mathcal{F}$	35.4	57.3	37.5
Shen *et al.* $_{\text{CVPR '19}}$ [189]	$\mathcal{I}$	6.1	11.7	5.5
Laradji *et al.* $_{\text{arXiv '19}}$ [190]	$\mathcal{I},\mathcal{P}$	7.8	18.2	8.8
Hsu *et al.* $_{\text{NeurIPS '19}}$ [102]	$\mathcal{B}$	21.1	45.5	17.2
BBAM (Ours)	$\mathcal{B}$	**26.0**	**50.0**	**23.9**
MS COCO *test-dev* images				
Mask R-CNN $_{\text{ICCV '17}}$ [20]	$\mathcal{F}$	35.7	58.0	37.8
Fan *et al.* $_{\text{ECCV '18}}$ [191]	$\mathcal{I},\mathcal{S}_I$	13.7	25.5	13.5
LIID $_{\text{TPAMI '20}}$ [95]	$\mathcal{I}$	16.0	27.1	16.5
BBAM (Ours)	$\mathcal{B}$	**25.7**	**50.0**	**23.3**

$\mathcal{F}$—Full, $\mathcal{I}$—Image label, $\mathcal{P}$—Point, $\mathcal{B}$—Box, $\mathcal{S}_I$—Instance saliency

5.4.3 Weakly Supervised Semantic Segmentation

Table 5.3 compares published mIoU values achieved by recent methods performing semantic segmentation on validation and test images from the PASCAL VOC 2012 dataset. Since the labels for test images are not publicly available, the results for the test images were obtained from the official PASCAL VOC evaluation server. Our method, using the BBAM, yields an mIoU value of 73.7 for both the validation and the test images in the PASCAL VOC 2012 semantic segmentation benchmark. Our method outperforms all the methods that use image-level tags or bounding boxes for supervision. This new state-of-the-art performance was achieved with vanilla DeepLab-v2 [18] without any modifications to networks or additional training techniques, such as label refinement during training [98], recursive training [24], or fine-tuning with

Table 5.3: Weakly supervised semantic segmentation on PASCAL VOC 2012 *val* and *test* images.

Method	*val*	*test*
Full supervision: Semantic masks		
DeepLab $_{\text{TPAMI '17}}$ [18]	76.8	76.2
Weak supervision: Image-level tags		
FickleNet $_{\text{CVPR '19}}$ [27]	64.9	65.3
CIAN $_{\text{AAAI '20}}$ [118]	64.3	65.3
Chang *et al.* $_{\text{CVPR '20}}$ [30]	66.1	65.9
Sun *et al.* $_{\text{ECCV '20}}$ [94]	66.2	66.9
Weak Supervision: Bounding boxes		
WSSL $_{\text{ICCV '15}}$ [192]	60.6	62.2
BoxSup $_{\text{ICCV '15}}$ [98]	62.0	64.6
SDI $_{\text{CVPR '17}}$ [24]	69.4	-
Song *et al.* $_{\text{CVPR '19}}$ [25]	70.2	-
BBAM (Ours)	**73.7**	**73.7**

additional losses [25]. Figure 5.3 presents examples of semantic masks produced by our method.

The concurrent method, Box2Seg [99], achieved an mIoU of 76.4% on the PASCAL VOC validation images, but it is based on UperNet [193], which is a more powerful segmentation network than DeepLab-v2 [150]. For a fair comparison between Box2Seg [99] and our BBAM, we attempt to relieve the benefit of UperNet [193] over DeepLab-v2 [18] by comparing the relative performance of the weakly supervised model to the fully supervised model. Box2Seg achieves 88.4% of the performance of its fully supervised equivalent (76.4 *vs.* 86.4); but the corresponding figure for BBAM and its fully supervised equivalent is 96.7% (73.7 *vs.* 76.2).

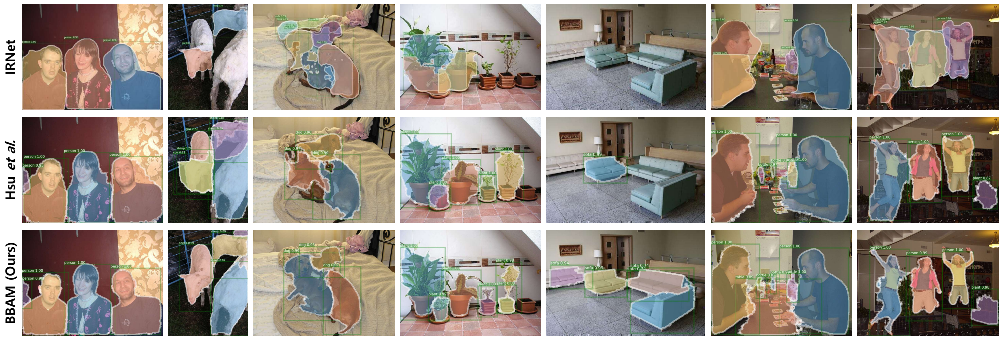


Figure 5.2: Examples of predicted instance masks for PASCAL VOC *val* images of IRNet [2], Hsu *et al.* [102], and ours.

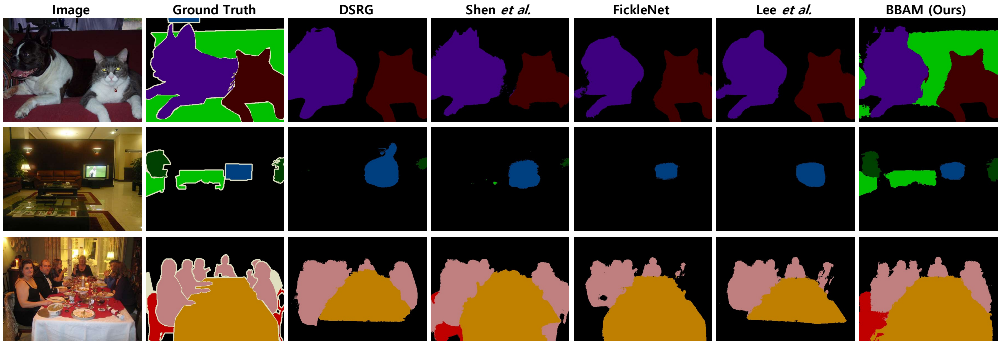


Figure 5.3: Examples of predicted semantic masks for PASCAL VOC *val* images of DSRG [91], Shen *et al.* [35], FickleNet [27], Lee *et al.* [33], and our method.

5.4.4 Ablation Study

MCG proposals. Table 5.4 shows how mask refinement with MCG proposals improves the instance segmentation performance of our method on the PASCAL VOC and MS COCO datasets. Mask refinement with MCG proposals is particularly effective on masks for medium and large objects. The results obtained without MCG proposals offer the possibility of a fairer comparison with Hsu *et al.* [102], which do not use MCG proposals. Our method produces better results than that of Hsu *et al.* [102] for both the PASCAL VOC and MS COCO datasets, which are shown in Tables 5.1 and 5.2 respectively. Hereinafter, to observe the contribution of each component of our system,

Table 5.4: Effectiveness of using MCG proposals for instance segmentation. AP_S, AP_M, and AP_L respectively denote the AP values for small, medium, and large objects.

MCG	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
PASCAL VOC *val* images:						
✗	29.6	61.9	25.8	5.6	21.6	40.1
✓	**33.4**	**63.7**	**31.8**	**6.5**	**26.4**	**44.1**
MS COCO *val* images:						
✗	23.5	47.9	20.3	10.4	24.9	36.5
✓	**26.0**	**50.0**	**23.9**	**10.8**	**28.5**	**40.3**

we report results without using MCG proposals.

***Box* and *cls heads*.** BBAM can provide a separate attribution map for each head of the object detector by controlling the logical variables $\mathbb{1}_{\text{box}}$ and $\mathbb{1}_{\text{cls}}$ in Eq. 5.3. Figure 5.4 shows the effect of the BBAM obtained from each head on the performance of weakly supervised semantic and instance segmentation. Using the BBAM obtained from either the *box head* ($\mathbb{1}_{\text{box}} = 1$ and $\mathbb{1}_{\text{cls}} = 0$) or the *cls head* ($\mathbb{1}_{\text{box}} = 0$ and $\mathbb{1}_{\text{cls}} = 1$) shows competent performance, but the best performance is achieved when the two heads are used together. We attribute this to the complementary property of the two heads, which is examined in more detail in Section 5.5.

Parameter sensitivity analysis. Table 5.5 shows the effect of the thresholds θ_{fg} and θ_{bg}, and the seed growing technique $\mathcal{G}$. When θ_{fg} equals to θ_{bg}, all pixels are assigned to either the foreground or the background. We see that ignoring some pixels can improve the AP values, and the seed growing technique further improves performance. We then studied the effect of λ, which controls the sparsity of the BBAM, on the performance of weakly supervised semantic and instance segmentation, with the results shown in Table 5.6. Our method shows similar performance on semantic and instance segmentation over a broad range of values of λ.

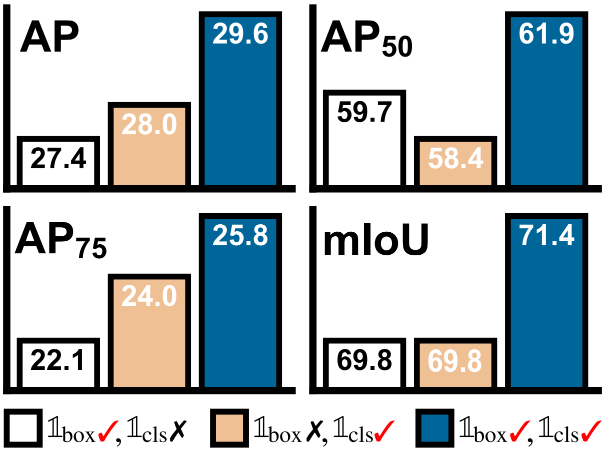


Figure 5.4: Effect of each head on instance and semantic segmentation.

5.5 Detailed Analysis of the BBAM

Examples of BBAMs. Figure 5.5 shows BBAMs for validation images from PASCAL VOC [32] and MS COCO [79]. The BBAMs have high values on the boundary and discriminative parts of each object, which are informative in conducting object detection.

Complementary operation of the *box* and *cls heads*. To determine which regions of an object are important to each head, we investigated the distribution of high-value pixels in the BBAM produced by each head. In Figure 5.6(a), $\mathcal{C}$ is the set of points on the contour of the object mask, and $\vec{x_c}$ is its centroid. For each pixel $\vec{x}$, we determine $r_1 = \|\vec{x} - \vec{x_c}\|_2$ and $r_2 = \min_{\vec{c}\in\mathcal{C}} \|\vec{x} - \vec{c}\|_2$. Letting the angle between $\vec{x} - \vec{x_c}$ and the x-axis be θ, the position of the pixel $\vec{x}$ relative to $\vec{x_c}$ is $\vec{R} = (\frac{r_1}{r_1+r_2}\cos\theta, \frac{r_1}{r_1+r_2}\sin\theta)$. In Figure 5.6(b), we plot the relative positions of all the pixels with attribution values above 0.9 obtained from validation images of the PASCAL VOC dataset. Pixels for which $\|\vec{R}\|_2 \approx 1$ are near the boundary of the object. We observed that high values attributed by the *box head* mainly occur near the boundary of the object, and those by the *cls head* mainly occur in the interior.

Furthermore, we observed how much the prediction of each head changes when either of $\mathbb{1}_{\text{box}}$ and $\mathbb{1}_{\text{cls}}$ is set to 1 during the optimization of Eq. 5.1. The extent of

θ_{fg}	θ_{bg}	$\mathcal{G}$	AP	AP$_{50}$	AP$_{75}$
0.2	0.2	✗	24.8	58.3	18.1
0.5	0.5	✗	28.3	59.5	24.7
0.8	0.8	✗	27.8	59.0	23.3
0.3	0.7	✗	28.1	59.5	24.0
0.3	0.7	✓	28.4	59.6	24.6
0.2	0.8	✗	28.6	60.4	24.0
0.2	0.8	✓	**29.6**	**61.9**	**25.8**

Table 5.5: Analysis of thresholds θ_{fg} and θ_{bg}, and effect of the growing technique $\mathcal{G}$.

		`Ins.`		`Sem.`
λ	AP	AP$_{50}$	AP$_{75}$	mIoU
0.001	26.6	58.7	21.1	67.9
0.003	28.1	59.9	22.8	69.7
0.005	28.7	60.2	24.3	70.8
0.007	**29.6**	**61.9**	**25.8**	**71.4**
0.010	28.7	60.4	24.4	70.7
0.020	28.3	59.6	23.7	70.3

Table 5.6: Effect of λ on instance (`Ins.`) and semantic (`Sem.`) segmentation.

the change in prediction of each head can be inferred from the corresponding loss in Eq. 5.2. Figure 5.6(c) shows that applying the optimization of Eq. 5.1 to one of the heads increases the loss of the other head, implying that the discriminative area of the image necessary for each head is not sufficient for the other head to maintain the prediction. These two observations suggest that the BBAM of each head provides complementary attributions. Examples of BBAMs obtained from each head are presented in the Appendix.

Label noise in object detection. We also looked at the robustness of our system against noisy box coordinate labels in instance segmentation. Hsu *et al.* [102] considered the effect of up to $\pm 15\%$ of label noise: we extend this to $\pm 20\%$. The validity of the bounding box tightness priors used by Hsu *et al.* [102] is seriously compromised by inaccurate box coordinates, with a considerable effect on performance, as shown in Figure 5.7(a). Our method shows better robustness than that of Hsu *et al.* [102], whether the noise consists of expanded or contracted bounding box annotations.

Effectiveness of an adaptive stride $s(a)$. As mentioned earlier, we use an adaptive

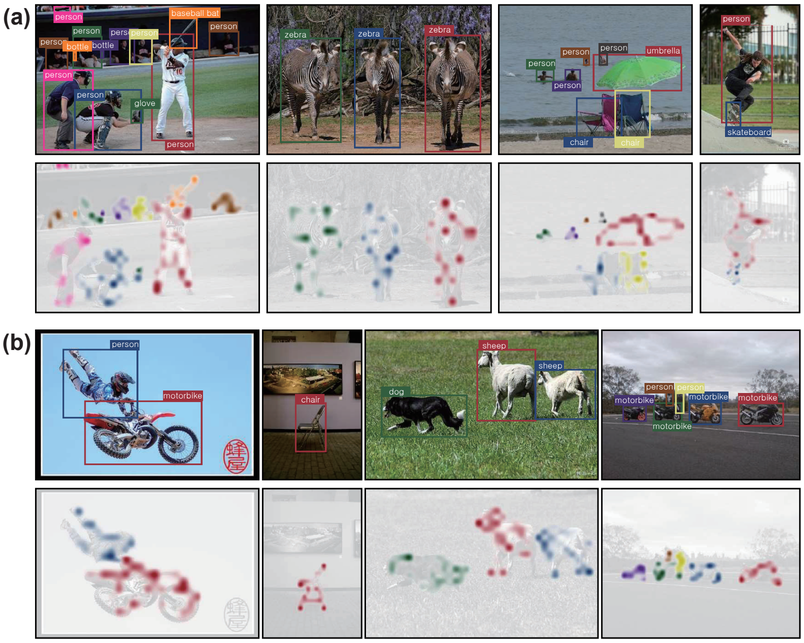


Figure 5.5: Examples of the predicted boxes and corresponding BBAMs. (a) BBAMs for MS COCO validation images. (b) BBAMs for PACSAL VOC validation images. Each BBAM corresponds to the predicted box of the same color.

stride $16 \leq s(a) \leq 64$ to cope with feature transformation due to RoI pooling. Figure 5.7(b) shows the IoU between the BBAM and ground truth mask on PASCAL VOC validation images, along with the results using fixed strides of 24 and 48. Figure 5.7(b) shows that a small fixed stride (s=24) is ineffective with large objects, as is a large fixed stride (s=48) with small objects. By contrast, an adaptive stride $s(a)$ can deal with objects of various sizes.

Comparison with gradient-based methods. Gradient-based attribution methods, such as SimpleGrad [181], SmoothGrad [182], and Grad-CAM [37] can also provide attributions for the output of an object detector. However, since only the subset of features associated with the imperfect proposal is delivered to the *cls* and *box* heads, the gradients with respect to pixels, which exist outside the proposal yet essential for prediction, can vanish (but not completely, due to the receptive field). We provide

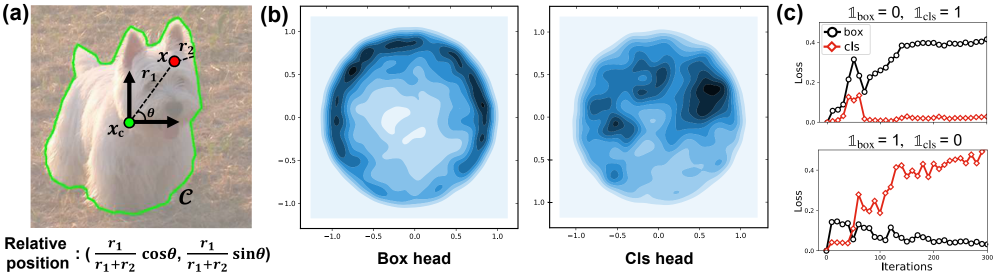


Figure 5.6: Complementary operation of the *box head* and the *cls head*. (a) The definition of relative position. (b) Relative positions of the highly activated pixels from each head. (c) *Box* and *class* loss curves.

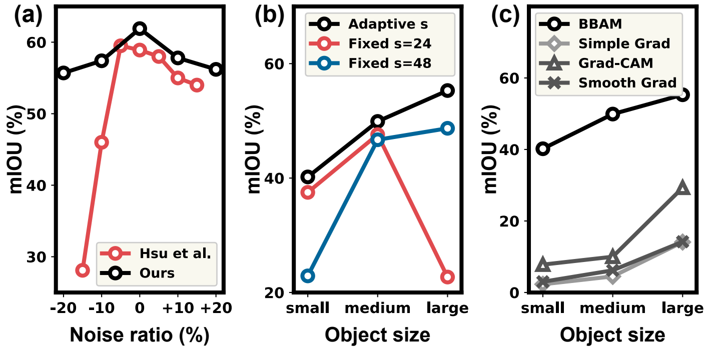


Figure 5.7: (a) Robustness against noisy box coordinate labels. (b) Localization accuracy by different strides. (c) Localization accuracy by different attribution methods.

empirical results supporting this analysis on the PASCAL VOC validation images: **(1)** Figure 5.8 shows examples in which SimpleGrad [181] is applied to three similar predictions from different proposals. Pixels outside the proposal do indeed influence the predictions, but SimpleGrad's attributions mainly appear inside the proposal. **(2)** We observed that the majority (87%) of pixels with attribution values above 0.9 appear inside the imperfect proposal; the mean IoU between the set of positive proposals and the corresponding predictions is low (*i.e.,* 0.56). **(3)** Figure 5.7(c) shows that attribution maps from gradient-based attribution methods correlate poorly with ground truth masks.

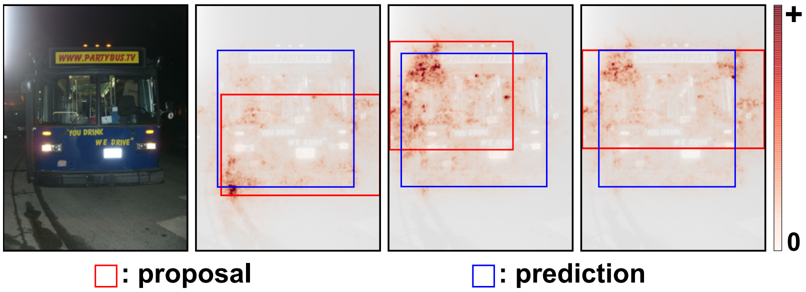


Figure 5.8: Examples of SimpleGrad [181] for three similar predictions obtained from different proposals.

5.6 Summary

We have introduced a bounding box attribution map (BBAM), which provides pixel-level localization of each target object in its bounding box by finding the smallest region that preserves the predictions of the object detector. Our formulation is built on two-stage object detectors, but applying our method to one-stage object detectors is straightforward as long as they have *box* and *cls* heads. Our experiments demonstrate that the BBAM achieves a new state-of-the-art on the PASCAL VOC and MS COCO benchmarks in weakly supervised semantic and instance segmentation. We have also analyzed BBAMs from various viewpoints, and compared our technique with other attribution methods, to provide a deeper understanding of our approach. We expect BBAMs to be a staple of future work on weakly supervised semantic and instance segmentation with bounding boxes, on a par with the CAM for class labels.

Chapter 6

Conclusion

In this dissertation, we have proposed various types of weak supervision for learning representative object recognition tasks, *i.e.,* semantic segmentation and instance segmentation. This chapter summarizes our contributions to label-efficient learning for object recognition and discusses future directions.

6.1 Dissertation Summary

Label-efficient learning of object recognition is one of the key factors for the successful utilization of deep neural networks into real-life applications. However, learning from weak supervision is not trivial. In this dissertation, we have addressed learning semantic and instance segmentation from various types of weak supervision. Figure 6.1 provides wrap-up summary of the motivations of utilizing different types of weak supervision. In Chapter 3, we have studied three methods for weakly supervised semantic segmentation using only image-level class labels. FickleNet chooses features at random during both training and inference, so that we obtain many different localization maps from a single image, and then aggregate those maps into a single localization map. AdvCAM manipulates images with a pixel-level perturbation, which is obtained from the gradient computed from the output of the classifier with respect to the input image, resulting in

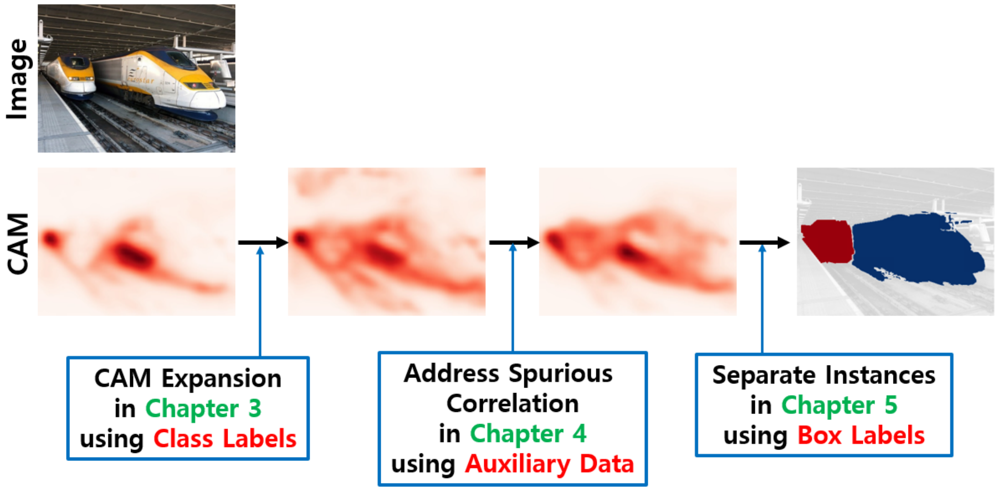


Image
CAM
CAM Expansion in Chapter 3 using Class Labels
Address Spurious Correlation in Chapter 4 using Auxiliary Data
Separate Instances in Chapter 5 using Box Labels

Figure 6.1: Wrap-up summary of the motivations of utilizing different types of weak supervision.

the increased classification score of the perturbed image. The attribution map of the manipulated image covers more of the target object. In RIB, through the information bottleneck principle, we first analyzed why the localization map obtained from a classifier identifies only a small region of the target object. Our analysis highlighted that the amount of information delivered from an input image to the output classification is largely determined by the final layer of the DNN. We then developed a method to reduce the information bottleneck through two simple modifications to the existing training scheme: the removal of the final non-linear activation function in the DNN and the introduction of a new pooling method.

The above methods significantly improved the performance of weakly supervised semantic segmentation using image-level class labels, but the spurious correlation problem cannot be addressed. The spurious correlation problem incurs confusion between foreground and background cues, resulting in an inaccurate segmentation map. Therefore, in Chapter 4, we have proposed the use of a new source of information, the out-of-distribution (OoD) data, for suppressing the spurious correlations. We have collected OoD data and proposed the metric learning based technique, W-OoD, to suppress the correlation between foreground and background.

Image-level class labels have led to significant achievements in semantic segmentation, but they are inherently unhelpful in instance segmentation, which requires the separation of different objects of the same class. In contrast, bounding boxes do provide information about the location of individual objects in an image. Therefore, in Chapter 5, we have studied the method of using bounding box labels for learning instance segmentation. In this work, we utilize higher-level information from the behavior of a trained object detector: Bounding Box Attribution Map (BBAM) is obtained by seeking the smallest areas of the image from which the object detector produces almost the same result as it does from the whole image.

6.2 Limitations and Future Direction

In this dissertation, we have focused only on a single type of weak supervision for each method. However, the promising way of reducing the total annotation cost is to use mixed types of supervision. In the real-world scenario, there may be several existing datasets containing classes of our interest, but those datasets may have different levels of supervision (*e.g.,* one has only class labels, but the other one contains bounding boxes). In order to maximize the performance of the trained segmentation model, we should be able to utilize a mixture of these various types of weak supervision. However, there are some stumbling blocks to utilizing mixed types of supervision.

To realize the training from the mixed types of weak supervision, we should design a holistic approach that can consider all the types of supervision in the same manner. Existing methods have assumed they have only a single form of weak supervision, and have designed methods specific to the given weak supervision. Therefore, a holistic method that can extract meaningful information from various types of weak supervision is required.

When designing the method of utilizing multiple datasets for training together, the data distribution shift issue should also be considered. Because each dataset has its own data distribution, training without considering the data distribution shift may have a negative effect on performance. The domain adaptation researches [194] or domain generalization researches [195] will be helpful for addressing the data distribution shift problem.

Bibliography

[1] J. Choe, S. Lee, and H. Shim, “Attention-based dropout layer for weakly supervised single object localization and semantic segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[2] J. Ahn, S. Cho, and S. Kwak, “Weakly supervised learning of instance segmentation with inter-pixel relations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.

[4] S. Nah, T. Hyun Kim, and K. M. Lee, “Deep multi-scale convolutional neural network for dynamic scene deblurring,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3883–3891.

[5] S. Nah, S. Son, and K. M. Lee, “Recurrent neural networks with intra-frame iterations for video deblurring,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8102–8111.

[6] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, “Enhanced deep residual networks for single image super-resolution,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144.

[7] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646–1654.

[8] J. Choi, J. Lee, Y. Jeong, and S. Yoon, "Toward spatially unbiased generative models," *arXiv preprint arXiv:2108.01285*, 2021.

[9] J. Choi, J. Lee, C. Shin, S. Kim, H. Kim, and S. Yoon, "Perception prioritized training of diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 472–11 481.

[10] J. Lee, J. Lee, S. Lee, and S. Yoon, "Mutual suppression network for video prediction using disentangled features," *arXiv preprint arXiv:1804.04810*, 2018.

[11] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[12] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4293–4302.

[13] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6023–6032.

[14] S. Yun, S. J. Oh, B. Heo, D. Han, J. Choe, and S. Chun, "Re-labeling imagenet: from single to multi-labels, from global to localized labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[17] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[18] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[19] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *European Conference on Computer Vision*, 2018.

[20] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[21] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, "What's the point: Semantic segmentation with point supervision," in *European Conference on Computer Vision*, 2016.

[22] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[23] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers, "Normalized cut loss for weakly-supervised cnn segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[24] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, “Simple does it: Weakly supervised instance and semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[25] C. Song, Y. Huang, W. Ouyang, and L. Wang, “Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[26] J. Lee, J. Yi, C. Shin, and S. Yoon, “Bbam: Bounding box attribution map for weakly supervised semantic and instance segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[27] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon, “Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[28] W. Shimoda and K. Yanai, “Self-supervised difference detection for weakly-supervised semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[29] J. Ahn and S. Kwak, “Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[30] Y.-T. Chang, Q. Wang, W.-C. Hung, R. Piramuthu, Y.-H. Tsai, and M.-H. Yang, “Weakly-supervised semantic segmentation via sub-category exploration,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[31] E. Kim, S. Kim, J. Lee, H. Kim, and S. Yoon, “Bridging the gap between classification and localization for weakly supervised object localization,” in

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14 258–14 267.

[32] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, 2010.

[33] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon, "Frame-to-frame aggregation of active regions in web videos for weakly supervised semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[34] S. Hong, D. Yeo, S. Kwak, H. Lee, and B. Han, "Weakly supervised semantic segmentation using web-crawled videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[35] T. Shen, G. Lin, C. Shen, and I. Reid, "Bootstrapping the performance of webly supervised semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[36] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[37] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[38] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *International Conference on Learning Representation*, 2014.

[39] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial machine learning at scale,” in *International Conference on Learning Representation*, 2017.

[40] N. Tishby and N. Zaslavsky, “Deep learning and the information bottleneck principle,” in *2015 IEEE Information Theory Workshop (ITW)*. IEEE, 2015, pp. 1–5.

[41] R. Shwartz-Ziv and N. Tishby, “Opening the black box of deep neural networks via information,” *arXiv preprint arXiv:1703.00810*, 2017.

[42] A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox, “On the information bottleneck theory of deep learning,” in *International Conference on Learning Representations*, 2018.

[43] N. Tishby, F. C. Pereira, and W. Bialek, “The information bottleneck method,” *arXiv preprint physics/0004057*, 2000.

[44] Y. Dubois, D. Kiela, D. J. Schwab, and R. Vedantam, “Learning optimal representations with the decodable information bottleneck,” in *Advances in Neural Information Processing Systems*, 2020.

[45] A. Achille and S. Soatto, “Information dropout: Learning optimal representations through noisy computation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[46] J. Choe, S. J. Oh, S. Lee, S. Chun, Z. Akata, and H. Shim, “Evaluating weakly supervised object localization methods right,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[47] K. Li, Z. Wu, K.-C. Peng, J. Ernst, and Y. Fu, “Guided attention inference network,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[48] D. Zhang, H. Zhang, J. Tang, X. Hua, and Q. Sun, “Causal intervention for weakly-supervised semantic segmentation,” in *Advances in Neural Information Processing Systems*, 2020.

[49] J. Lee, J. Choi, J. Mok, and S. Yoon, “Reducing information bottleneck for weakly supervised semantic segmentation,” in *Advances in Neural Information Processing Systems*, 2021.

[50] S. Lee, M. Lee, J. Lee, and H. Shim, “Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[51] A. Kolesnikov and C. H. Lampert, “Improving weakly-supervised object localization by micro-annotation,” in *British Machine Vision Conference*, 2016.

[52] Y. Yao, T. Chen, G. Xie, C. Zhang, F. Shen, Q. Wu, Z. Tang, and J. Zhang, “Non-salient region object mining for weakly supervised semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[53] S. Kwak, S. Hong, and B. Han, “Weakly supervised semantic segmentation using superpixel pooling network,” in *The Association for the Advancement of Artificial Intelligence*, 2017.

[54] X. Wang, S. You, X. Li, and H. Ma, “Weakly-supervised semantic segmentation by iteratively mining common object features,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[55] M. Bellver, A. Salvador, J. Torrres, and X. Giro-i Nieto, “Budget-aware semi-supervised semantic and instance segmentation,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[56] C. Rother, V. Kolmogorov, and A. Blake, “” grabcut” interactive foreground extraction using iterated graph cuts,” *ACM transactions on graphics (TOG)*, 2004.

[57] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, and J. Malik, “Multiscale combinatorial grouping for image segmentation and object proposal generation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[58] S.-g. Lee, J. S. Bae, H. Kim, J. H. Kim, and S. Yoon, “Liver lesion detection from weakly-labeled multi-phase ct volumes with a grouped single shot multibox detector,” in *MICCAI*, 2018.

[59] S. Lee, J. Lee, J. Lee, C.-K. Park, and S. Yoon, “Robust tumor localization with pyramid grad-cam,” *arXiv preprint arXiv:1805.11393*, 2018.

[60] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, “Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6629–6638.

[61] A. Padmakumar, J. Thomason, A. Shrivastava, P. Lange, A. Narayan-Chen, S. Gella, R. Piramuthu, G. Tur, and D. Hakkani-Tur, “Teach: Task-driven embodied agents that chat,” in *Proceedings of the The Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, vol. 36, no. 2, 2022, pp. 2017–2025.

[62] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition,” *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[63] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2015.

[64] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[65] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[66] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[67] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," *arXiv preprint arXiv:2203.03605*, 2022.

[68] H. Song, D. Sun, S. Chun, V. Jampani, D. Han, B. Heo, W. Kim, and M.-H. Yang, "Vidt: An efficient and effective fully transformer-based object detector," *arXiv preprint arXiv:2110.03921*, 2021.

[69] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[70] R. Nock and F. Nielsen, "Statistical region merging," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 26, no. 11, pp. 1452–1458, 2004.

[71] N. Dhanachandra, K. Manglem, and Y. J. Chanu, "Image segmentation using k-means clustering algorithm and subtractive clustering algorithm," *Procedia Computer Science*, vol. 54, pp. 764–771, 2015.

[72] S. Vicente, V. Kolmogorov, and C. Rother, “Graph cut based image segmentation with connectivity priors,” in *2008 IEEE conference on computer vision and pattern recognition*. IEEE, 2008, pp. 1–8.

[73] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[74] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[75] P. Krähenbühl and V. Koltun, “Efficient inference in fully connected crfs with gaussian edge potentials,” in *Advances in Neural Information Processing Systems*, 2011.

[76] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, “Ccnet: Criss-cross attention for semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[77] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, “Attention to scale: Scale-aware semantic image segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3640–3649.

[78] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, “Mask scoring r-cnn,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6409–6418.

[79] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European Conference on Computer Vision*, 2014.

[80] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “Yolact: Real-time instance segmentation,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9157–9166.

[81] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, “Solo: Segmenting objects by locations,” in *European Conference on Computer Vision*. Springer, 2020, pp. 649–665.

[82] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother, “Instancecut: from edges to instances with multicut,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5008–5017.

[83] M. Bai and R. Urtasun, “Deep watershed transform for instance segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5221–5229.

[84] A. Arnab and P. H. Torr, “Pixelwise instance segmentation with a dynamically instantiated network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 441–450.

[85] J. Fan, Z. Zhang, C. Song, and T. Tan, “Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[86] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, “Object region mining with adversarial erasing: A simple classification to semantic segmentation approach,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1568–1576.

[87] K. Li, Z. Wu, K.-C. Peng, J. Ernst, and Y. Fu, “Tell me where to look: Guided attention inference network,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[88] K. K. Singh and Y. J. Lee, “Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[89] X. Zhang, Y. Wei, J. Feng, Y. Yang, and T. S. Huang, “Adversarial complementary learning for weakly supervised object localization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[90] Q. Hou, P. Jiang, Y. Wei, and M.-M. Cheng, “Self-erasing network for integral object attention,” in *Advances in Neural Information Processing Systems*, 2018.

[91] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, “Weakly-supervised semantic segmentation network with deep seeded region growing,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[92] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, “The secrets of salient object segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014.

[93] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, “Learning to detect a salient object,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[94] G. Sun, W. Wang, J. Dai, and L. Van Gool, “Mining cross-image semantics for weakly supervised semantic segmentation,” in *European Conference on Computer Vision*, 2020.

[95] Y. Liu, Y.-H. Wu, P.-S. Wen, Y.-J. Shi, Y. Qiu, and M.-M. Cheng, “Leveraging instance-, image-and dataset-level information for weakly supervised instance segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[96] S. Joon Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele, "Exploiting saliency for object segmentation from image level labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[97] T.-W. Ke, J.-J. Hwang, and S. X. Yu, "Universal weakly supervised segmentation by pixel-to-segment contrastive learning," in *International Conference on Learning Representation*, 2021.

[98] J. Dai, K. He, and J. Sun, "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2015.

[99] V. Kulharia, S. Chandra, A. Agrawal, P. Torr, and A. Tyagi, "Box2seg: Attention weighted loss and discriminative feature learning for weakly supervised segmentation," in *European Conference on Computer Vision*, 2020.

[100] S. Liao, Y. Sun, C. Gao, P. S. KP, S. Mu, J. Shimamura, and A. Sagata, "Weakly supervised instance segmentation using hybrid networks," in *ICASSP*, 2019.

[101] Y. Sun, S. Liao, C. Gao, C. Xie, F. Yang, Y. Zhao, and A. Sagata, "Weakly supervised instance segmentation based on two-stage transfer learning," *IEEE Access*, 2020.

[102] C.-C. Hsu, K.-J. Hsu, C.-C. Tsai, Y.-Y. Lin, and Y.-Y. Chuang, "Weakly supervised instance segmentation using the bounding box tightness prior," in *Advances in Neural Information Processing Systems*, 2019.

[103] A. Arun, C. Jawahar, and M. P. Kumar, "Weakly supervised instance segmentation by learning annotation consistent instances," in *European Conference on Computer Vision*, 2020.

[104] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional

networks," in *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2018, pp. 839–847.

[105] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 24–25.

[106] J. R. Lee, S. Kim, I. Park, T. Eo, and D. Hwang, "Relevance-cam: Your model already knows where to look," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 944–14 953.

[107] Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen, "Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[108] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang, "Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[109] W. Bae, J. Noh, and G. Kim, "Rethinking class activation mapping for weakly supervised object localization," in *European Conference on Computer Vision*. Springer, 2020, pp. 618–634.

[110] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2083–2090.

[111] M. Lee, D. Kim, and H. Shim, "Threshold matters in wsss: Manipulating the activation for the robust and accurate segmentation model against thresholds,"

in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4330–4339.

[112] J. Lee, E. Kim, and S. Yoon, "Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[113] J. Lee, E. Kim, J. Mok, and S. Yoon, "Anti-adversarially manipulated attributions for weakly supervised semantic segmentation and object localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[114] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[115] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[116] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[117] D. Kim, D. Cho, D. Yoo, and I. S. Kweon, "Two-phase learning for weakly supervised object localization," *arXiv preprint arXiv:1708.02108*, 2017.

[118] J. Fan, Z. Zhang, and T. Tan, "Cian: Cross-image affinity net for weakly supervised semantic segmentation," *The Association for the Advancement of Artificial Intelligence*, 2020.

[119] A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: Three principles for weakly-supervised image segmentation," in *European Conference on Computer Vision*, 2016.

[120] L. Chen, W. Wu, C. Fu, X. Han, and Y. Zhang, “Weakly supervised semantic segmentation with boundary exploration,” in *European Conference on Computer Vision*, 2020.

[121] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “Deepfool: a simple and accurate method to fool deep neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[122] A. Arnab, O. Miksik, and P. H. Torr, “On the robustness of semantic segmentation models to adversarial attacks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[123] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, “Adversarial examples for semantic segmentation and object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[124] A.-K. Dombrowski, M. Alber, C. Anders, M. Ackermann, K.-R. Müller, and P. Kessel, “Explanations can be manipulated and geometry is to blame,” in *Advances in Neural Information Processing Systems*, 2019.

[125] J. Heo, S. Joo, and T. Moon, “Fooling neural network interpretations via adversarial model manipulation,” in *Advances in Neural Information Processing Systems*, 2019.

[126] S.-M. Moosavi-Dezfooli, A. Fawzi, J. Uesato, and P. Frossard, “Robustness via curvature regularization, and vice versa,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[127] C. Qin, J. Martens, S. Gowal, D. Krishnan, K. Dvijotham, A. Fawzi, S. De, R. Stanforth, and P. Kohli, “Adversarial robustness through local linearization,” in *Advances in Neural Information Processing Systems*, 2019.

[128] S. Santurkar, A. Ilyas, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Image synthesis with a single (robust) classifier," in *Advances in Neural Information Processing Systems*, 2019.

[129] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," in *International Conference on Learning Representation*, 2019.

[130] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Advances in Neural Information Processing Systems*, 2019.

[131] W. Liu, C. Zhang, G. Lin, T.-Y. HUNG, and C. Miao, "Weakly supervised segmentation with maximum bipartite graph matching," in *ACMMM*, 2020.

[132] Y. Ouali, C. Hudelot, and M. Tami, "Semi-supervised semantic segmentation with cross-consistency training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[133] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2011.

[134] W. Luo and M. Yang, "Semi-supervised semantic segmentation via strong-weak dual-branch network," in *European Conference on Computer Vision*, 2020.

[135] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[136] Z. Wu, C. Shen, and A. Van Den Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *Pattern Recognition*, 2019.

[137] K. Li, Y. Zhang, K. Li, Y. Li, and Y. Fu, "Attention bridging network for knowledge transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[138] T. Zhang, G. Lin, W. Liu, J. Cai, and A. Kot, "Splitting vs. merging: Mining object regions with discrepancy and intersection loss for weakly supervised semantic segmentation," in *European Conference on Computer Vision*, 2020.

[139] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a dcnn for semantic image segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2015.

[140] N. Souly, C. Spampinato, and M. Shah, "Semi supervised semantic segmentation using generative adversarial network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[141] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, 2008.

[142] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[143] B. Chen, W. Deng, and J. Du, "Noisy softmax: Improving the generalization ability of dcnn via postponing the early softmax saturation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[144] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[145] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[146] N. Araslanov and S. Roth, “Single-stage semantic segmentation from image labels,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[147] P. O. Pinheiro and R. Collobert, “From image-level to pixel-level labeling with convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1713–1721.

[148] Y.-T. Chang, Q. Wang, W.-C. Hung, R. Piramuthu, Y.-H. Tsai, and M.-H. Yang, “Mixup-cam: Weakly-supervised semantic segmentation via uncertainty regularization,” in *British Machine Vision Conference*, 2020.

[149] X. Li, T. Zhou, J. Li, Y. Zhou, and Z. Zhang, “Group-wise semantic mining for weakly supervised semantic segmentation,” in *The Association for the Advancement of Artificial Intelligence*, 2021.

[150] K. Nakashima, “DeepLab with PyTorch,” https://github.com/kazuto1011/deeplab-pytorch.

[151] B. Zhang, J. Xiao, Y. Wei, M. Sun, and K. Huang, “Reliability does matter: An end-to-end weakly supervised semantic segmentation approach,” *The Association for the Advancement of Artificial Intelligence*, 2020.

[152] J. Fan, Z. Zhang, and T. Tan, “Employing multi-estimations for weakly-supervised semantic segmentation,” in *European Conference on Computer Vision*, 2020.

[153] Q. Yao and X. Gong, “Saliency guided self-attention network for weakly and semi-supervised semantic segmentation,” *IEEE Access*, 2020.

[154] B. Kim, Y. Yoo, C. Rhee, and J. Kim, “Beyond semantic to instance segmentation: Weakly-supervised instance segmentation via semantic knowledge transfer and self-refinement,” *arXiv preprint arXiv:2109.09477*, 2021.

[155] A. Gupta, P. Dollar, and R. Girshick, “LVIS: A dataset for large vocabulary instance segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[156] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov *et al.*, “The open images dataset v4,” *International Journal of Computer Vision*, 2020.

[157] Y. Su, R. Sun, G. Lin, and Q. Wu, “Context decoupling augmentation for weakly supervised semantic segmentation,” *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[158] B. Jin, M. V. Ortiz Segovia, and S. Susstrunk, “Webly supervised semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[159] D. R. Vilar and C. A. Perez, “Extracting structured supervision from captions for weakly supervised semantic segmentation,” *IEEE Access*, 2021.

[160] J. Sawatzky, D. Banerjee, and J. Gall, “Harvesting information from captions for weakly supervised semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop*, 2019.

[161] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, “Learning to detect salient objects with image-level supervision,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[162] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, “Global contrast based salient region detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[163] T. Wu, J. Huang, G. Gao, X. Wei, X. Wei, X. Luo, and C. H. Liu, “Embedded discriminative attention mechanism for weakly supervised semantic segmentation,”

in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[164] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *International Conference on Learning Representation*, 2019.

[165] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," in *International Conference on Learning Representation*, 2018.

[166] S. Lee, C. Park, H. Lee, J. Yi, J. Lee, and S. Yoon, "Removing undesirable feature contributions using out-of-distribution data," in *International Conference on Learning Representation*, 2021.

[167] H. Kweon, S.-H. Yoon, H. Kim, D. Park, and K.-J. Yoon, "Unlocking the potential of ordinary classifier: Class-specific adversarial erasing framework for weakly supervised semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[168] Y. Li, Z. Kuang, L. Liu, Y. Chen, and W. Zhang, "Pseudo-mask matters in weakly-supervised semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[169] B. Zhang, J. Xiao, J. Jiao, Y. Wei, and Y. Zhao, "Affinity attention graph neural network for weakly supervised semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[170] L. Xu, W. Ouyang, M. Bennamoun, F. Boussaid, F. Sohel, and D. Xu, "Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[171] J. Lee, S. J. Oh, S. Yun, J. Choe, E. Kim, and S. Yoon, “Weakly supervised semantic segmentation using out-of-distribution data,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 897–16 906.

[172] R. C. Fong and A. Vedaldi, “Interpretable explanations of black boxes by meaningful perturbation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[173] R. Fong, M. Patrick, and A. Vedaldi, “Understanding deep networks via extremal perturbations and smooth masks,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[174] P. Dabkowski and Y. Gal, “Real time image saliency for black box classifiers,” in *Advances in Neural Information Processing Systems*, 2017.

[175] P.-T. Jiang, Q. Hou, Y. Cao, M.-M. Cheng, Y. Wei, and H. Xiong, “Integral object mining via online attention accumulation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[176] C.-H. Chang, E. Creager, A. Goldenberg, and D. Duvenaud, “Explaining image classifiers by counterfactual generation,” *International Conference on Learning Representation*, 2019.

[177] K. Schulz, L. Sixt, F. Tombari, and T. Landgraf, “Restricting the flow: Information bottlenecks for attribution,” *International Conference on Learning Representation*, 2020.

[178] L. Hoyer, M. Munoz, P. Katiyar, A. Khoreva, and V. Fischer, “Grid saliency for context explanations of semantic segmentation,” in *Advances in Neural Information Processing Systems*, 2019.

[179] E. Reif, A. Yuan, M. Wattenberg, F. B. Viegas, A. Coenen, A. Pearce, and B. Kim, “Visualizing and measuring the geometry of bert,” in *Advances in Neural Information Processing Systems*, 2019.

[180] T. Wu and X. Song, “Towards interpretable object detection by unfolding latent structures,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[181] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *European Conference on Computer Vision*, 2014.

[182] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, “Smoothgrad: removing noise by adding noise,” *arXiv preprint arXiv:1706.03825*, 2017.

[183] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *European Conference on Computer Vision*, 2016.

[184] Y. Zhou, Y. Zhu, Q. Ye, Q. Qiu, and J. Jiao, “Weakly supervised instance segmentation using class peak response,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[185] Y. Zhu, Y. Zhou, H. Xu, Q. Ye, D. Doermann, and J. Jiao, “Learning instance activation maps for weakly supervised instance segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[186] F. Massa and R. Girshick, “maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch,” https://github.com/facebookresearch/maskrcnn-benchmark, 2018.

[187] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representation*, 2015.

[188] J. Dai, K. He, and J. Sun, “Instance-aware semantic segmentation via multi-task network cascades,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[189] Y. Shen, R. Ji, Y. Wang, Y. Wu, and L. Cao, “Cyclic guidance for weakly supervised joint detection and segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[190] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vázquez, and M. Schmidt, “Instance segmentation with point supervision,” *arXiv preprint arXiv:1906.06392*, 2019.

[191] R. Fan, Q. Hou, M.-M. Cheng, G. Yu, R. R. Martin, and S.-M. Hu, “Associating inter-image salient instances for weakly supervised semantic segmentation,” in *European Conference on Computer Vision*, 2018.

[192] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, “Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2015.

[193] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, “Unified perceptual parsing for scene understanding,” in *Proceedings of the European Conference on Computer Vision (European Conference on Computer Vision)*, 2018, pp. 418–434.

[194] S. Lee, D. Kim, N. Kim, and S.-G. Jeong, “Drop to adapt: Learning discriminative features for unsupervised domain adaptation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 91–100.

[195] S. Seo, Y. Suh, D. Kim, G. Kim, J. Han, and B. Han, “Learning to optimize domain specific normalization for domain generalization,” in *European Conference on Computer Vision*. Springer, 2020, pp. 68–83.

초 록

딥러닝의 발전은 이미지 물체 인식 분야를 크게 발전시켰다. 하지만 이러한 발전은 수많은 학습 이미지와 각 이미지에 사람이 직접 생성한 물체의 위치 정보에 대한 레이블 덕분에 가능한 것이였다. 이미지 물체 인식 분야를 실생활에서 활용하기 위해서는 다양한 물체의 카테고리를 인식 할 수 있어야 하며, 이를 위해선 각 카테고리당 수많은 학습 데이터가 필요하다. 하지만 각 이미지당 물체의 위치를 각 픽셀마다 주석을 다는 것은 많은 비용이 들어간다. 이러한 정보를 얻을 때 필요한 비용은 약한지도학습으로 줄일 수 있다. 약한 지도 학습이란, 물체의 명시적인 위치 정보를 포함하는 레이블보다 더 값싸게 얻을 수는 있지만, 약한 위치 정보를 활용하여 뉴럴네트워크를 학습하는 것이다. 본 학위논문에서는 물체의 카테고리 정보, 학습 외 분포 데이터 (out-of-distribution) 데이터, 그리고 물체의 박스 레이블을 활용하는 약한지도학습 방법론들을 다룬다.

첫 번째로, 물체의 카테고리 정보를 이용한 약한 지도 학습을 다룬다. 대부분의 카테로기 정보를 활용하는 방법들은 학습된 분류기로부터 얻어진 기여도맵 (attribution map) 을 활용하지만, 이들은 물체의 일부만을 찾아내는 문제가 있다. 우리는 이 문제에 대한 근본 원인을 이론적인 관점에서 의논하고, 이 문제를 해결할 수 있는 세 가지의 방법론을 제안한다. 하지만, 물체의 카테고리 정보만 활용하게 되면 이미지의 전경과 배경이 악의적인 상관관계를 가진다고 잘 알려져 있다. 우리는 이러한 상관관계를 학습 외 분포 데이터를 활용하여 완화한다. 마지막으로, 물체의 카테고리 정보에 기반한 방법론들은 같은 카테고리의 다른 물체를 분리하지 못하기 때문에 인스턴스 분할 (instance segmentation) 에 적용되기는 힘들다. 따라서 물체의 박스 레이블을 활용한 약한 지도학습 방법론을 제안한다.

제안된 방법론을 통해 레이블을 제작하는 시간을 획기적으로 줄일 수 있다는 것을 실험결과를 통해 확인했다. 어려운 데이터셋인 Pascal VOC 에 대해 우리는 91%의 데이터 비용을 감소하면서, 강한 레이블로 학습된 비교군의 89%의 성능을 달성하였다. 또한, 물체의 박스 정보를 활용해서는 83% 의 데이터 비용을 감소하면서, 강한 레이블로 학습된 비교군의 96%의 성능을 달성하였다. 본 학위논문에서 제안된 방법론들이 딥러닝 기반의 물체 인식이 다양한 데이터와 다양한 환경에서 활용되는 데에 있어 도움이 되기를 기대한다.

주요어: 딥러닝, 컴퓨터 비전, 머신 러닝, 약한지도학습, 객체 인식, 세그멘테이션, 의미론적 객체분할

학번: 2017-29653