PRIOR KNOWLEDGE GUIDED WEAKLY SUPERVISED OBJECT DETECTION AND SEMANTIC SEGMENTATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

FATIH BALTACI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

FEBRUARY 2022

Approval of the thesis:

PRIOR KNOWLEDGE GUIDED WEAKLY SUPERVISED OBJECT DETECTION AND SEMANTIC SEGMENTATION

submitted by **FATIH BALTACI** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _____

Assist. Prof. Dr. Ramazan Gökberk Cinbiş
Supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Assoc. Prof. Dr. Sinan Kalkan
Computer Engineering, METU _____

Assist. Prof. Dr. Ramazan Gökberk Cinbiş
Computer Engineering, METU _____

Prof. Dr. Selim Aksoy
Computer Engineering, Bilkent University _____

Date: 10.02.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Fatih Baltacı

Signature :

ABSTRACT

PRIOR KNOWLEDGE GUIDED WEAKLY SUPERVISED OBJECT DETECTION AND SEMANTIC SEGMENTATION

Baltacı, Fatih
M.S., Department of Computer Engineering
Supervisor: Assist. Prof. Dr. Ramazan Gökberk Cinbiş

February 2022, 85 pages

State-of-the-art recognition models in computer vision are trained using annotated training data. Collecting manual annotation for images is a time-consuming and tedious task. Annotation time and difficulty also change across computer vision tasks. For example, object detection tasks require bounding-box annotations, which can be difficult to annotate, particularly in complex scenes, and semantic segmentation tasks require pixel-level annotations, which by definition requires a great amount of effort. Weakly-supervised learning methods, typically studied for object detection and semantic segmentation, aim to avoid such detailed annotations and instead rely on image-level labels indicating the presence or absence of object categories. Existing results, however, indicate that weakly-supervised learning methods tend to result in recognition models that significantly underperform their fully-supervised counterparts. To this end, towards reducing the performance gap between the weakly supervised and fully supervised approaches, this thesis explores the utilization of prior semantic knowledge about object categories in improving the weakly supervised training processes. We inject prior knowledge for object categories represented in terms of attributes or language-based class embeddings into existing weakly-supervised object

detection and semantic segmentation training approaches. Our experimental results show that the proposed method can clearly improve the recognition performance in several cases on benchmark datasets.

Keywords: weakly-supervised object detection, weakly-supervised semantic segmentation, prior knowledge guidance in machine learning

ÖZ

ÖN BİLGİ YÖNLENDİRMELİ ZAYIF GÖZETİMLİ NESNE TESPİTİ VE ANLAMSAL BÖLÜTLEME

Baltacı, Fatih
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Dr. Öğr. Üyesi. Ramazan Gökberk Cinbiş

Şubat 2022 , 85 sayfa

Bilgisayarlı görü alanındaki en gelişmiş tanıma modelleri, etiketli eğitim verileri kullanılarak eğitilmektedir. Görüntülerde el ile etiketleme yapmak genellikle zaman alan ve zorlu bir işlemdir. Etiketleme süresi ve zorluğu da bilgisayarlı görü görevlerine göre değişir. Örneğin, nesne tespit problemi, özellikle karmaşık sahnelerde etiketlemesi zor olabilen sınırlayıcı kutu etiketleri gerektirir. Anlamsal bölütleme ise, tanımı gereği büyük miktarda çaba gerektiren piksel düzeyinde etiketler gerektirmektedir. Tipik olarak nesne algılama ve anlamsal bölütleme için çalışılan zayıf denetimli öğrenme yöntemleri, bu tür ayrıntılı etiketlerden kaçınmayı ve bunun yerine nesne kategorilerinin varlığını veya yokluğunu gösteren görüntü düzeyinde etiketleri kullanmayı amaçlar. Bununla birlikte, mevcut sonuçlar, zayıf denetimli öğrenme yöntemlerinin, tam denetimli öğrenme yöntemlerine kıyasla önemli ölçüde düşük performans gösterme eğiliminde olduğunu göstermektedir. Bu tezde, zayıf denetimli ve tam denetimli yaklaşımlar arasındaki performans farkını azaltmaya yönelik olarak zayıf denetimli eğitimde nesne kategorileri hakkında anlamsal ön bilgilerden yararlanmayı amaçlamaktayız. Öznitelikler veya dil tabanlı nesne kategorileri için anlamsal ön bil-

gileri, mevcut zayıf denetimli nesne algılama ve anlamsal bölütleme eğitim yaklaşımlarına dahil etmekteyiz. Deneysel sonuçlarımız, önerilen yöntemin, standart veri kümelerinde çeşitli durumlarda tanıma performansını açıkça iyileştirebileceğini göstermektedir.

Anahtar Kelimeler: zayıf gözetimli nesne algılama, zayıf gözetimli anlamsal bölütleme, ön bilgi yönlendirmeli makine öğrenmesi

To my family.

ACKNOWLEDGMENTS

Firstly, I would like to thank my supervisor Assist. Prof. Dr. Ramazan Gökberk Cinbiş for all the guidance and motivation I received during this period. It was not possible to complete this thesis without his help.

I would like to thank the members of my thesis committee, Assoc. Prof. Sinan Kalkan and Prof. Selim Aksoy for their time and valuable comments.

I want to thank my colleague Abdullah Dursun for his technical support and advice during this process. As a result of our work together, we achieved satisfactory results for this thesis.

I want to thank my business partners Kürşat Aktaş and Onur Çakmak for their incredible support and patience in completing this thesis during this process.

I would like to thank my girlfriend Buse Balcı for her love, patience and support throughout my research and writing of this thesis.

Finally, I must express my deepest gratitude to my parents for their continued support. This success would not have been possible without them.

The work in this thesis was supported in part by the TUBITAK Grant 116E445. The numerical calculations reported in this thesis were partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

FS	Fully Supervised
WS	Weakly Supervised
FSOD	Fully Supervised Object Detection
WSOD	Weakly Supervised Object Detection
WSSS	Weakly Supervised Semantic Segmentation
WSL	Weakly Supervised Learning
PCL	Proposal Cluster Learning
MIL	Multiple Instance Learning
CFR	Conditional Random Field
DPM	Discriminatively Trained Part-based Model
MIST	Multiple Instance Self Training
AP	Average Precision
mAP	Mean Average Precision
IoU	Intersection over Union
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
AUC	Area Under Curve
WSDDN	Weakly Supervised Deep Detection Networks
aPY	Attribute Pascal and Yahoo
GloVe	Global Vectors for Word Representation
COCO	Common Objects in Context
VOC	Visual Object Classes

CAM	Class Activation Map

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

There are various well-known computer vision problems, such as object detection, autonomous driving, defect detection, medical imaging, person counting, semantic segmentation. New models are constantly being proposed to improve the models' performance. Following the success of CNNs (Convolutional Neural Networks), classification, object detection, and segmentation methods started to gain impressive success. Particularly, in object detection, single-stage models [17, 18, 19, 20, 21, 22] and two-stage models [23, 24, 25, 26, 27, 28] show promising performance with fully-supervised training. Similarly, semantic segmentation methods [29, 30, 12, 31] are used for fine-grained pixel-level predictions, and the state-of-the-are models rely on fully-supervised training settings.

Various fully-annotated datasets have been proposed in the literature, such as [8, 3, 4, 2], to improve the training and evaluation of the state-of-the-art in object detection and semantic segmentation. While these datasets have been greatly influential in advancing the approaches, it is inherently difficult to collect similar mid-scale or large-scale annotated datasets in real-world scenarios where the public datasets are insufficient. For example, the images themselves, as well as image-level labels, are required for classification tasks, instance-level bounding box annotations are required for detection tasks, and pixel-level labels are needed for segmentation tasks to be able to utilize the fully-supervised state-of-the-art. The difficulty of collecting such datasets can be observed through the annotation examples shown in Figure 1.1.

It is well-known that in most practical scenarios, training accurate and reliable deep

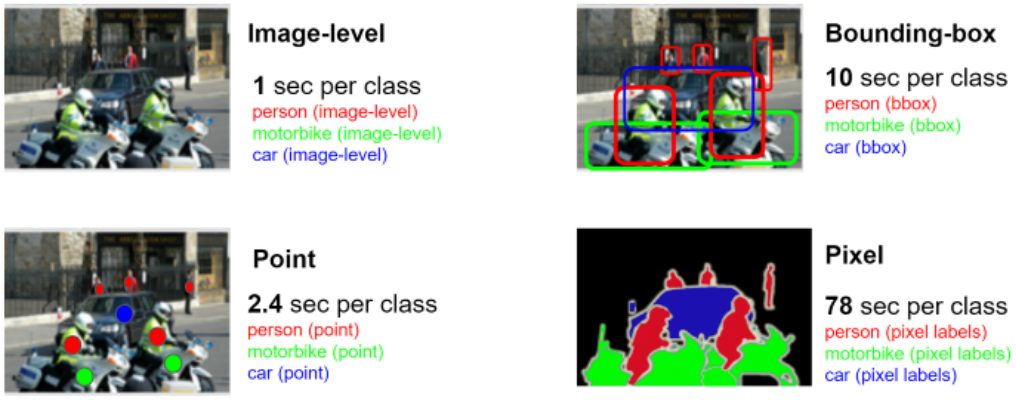


Figure 1.1: Annotating image-level labels is cheaper than annotating bounding-box, point of pixel labels. Pixel-level label annotation takes $\sim$ 78 sec per class for an image [32].

learning models typically require large-scale fully-annotated datasets. To collect datasets, therefore, careful data annotation is required, which is a labor-intensive and tedious task. To generate large-scale datasets, some public datasets [3, 33] have utilized Amazon Mechanical Turk [1] and similar interfaces to obtain distributed workforce to annotate images. While the crowd-sourcing strategy can appear appealing, its costs in large-scale settings can be prohibitive.

As it can visually be observed from the examples given Figure 1.1, the difficulty of collecting annotations can vary across the annotation types. For example, annotating an image can take around 10 seconds per instance for bounding boxes, 78 seconds per instance for pixel-wise segmentation annotations, few seconds per class for image-level labels [32]. Therefore, the cost of collecting image-wise labels is typically lower. In addition, in special cases, one can find alternative practical ways to collect images with specific labels, such as using image search engines combined with simple image cleaning.

Motivated from the relative simplicity of collecting image-wise labels, weakly-supervised object detection aims to build detection models from training images with only object

[1] https://www.mturk.com/

presence/absence annotations. There are various approaches for weakly-supervised object detection problems. A commonly used recent trend is to use MIL (Multiple Instance Learning) [34, 35, 36, 1, 37, 38, 39, 40, 41] to solve weakly-supervised object detection problems. Images and object proposals are interpreted as bags and instances, respectively, in MIL.

Weakly-supervised semantic segmentation is a similar problem that aims to train semantic segmentation models, as opposed to object detectors, via class-level annotated images. A commonly used recent trend is to use CAM (Class Activation Maps) [42] or similar techniques to generate pseudo-ground-truth pixel maps [9, 43, 15, 44, 45, 46, 47]. These pseudo-ground-truth maps are then used for semantic segmentation training. There exist also other supervision approaches for weakly-supervised semantic segmentation such as bounding box [48, 49], scribbles [50], and points [32] for training models. In this thesis, we focus on image-level labels as the weak supervision.

1.2 Contributions and Overview

Weakly-supervised training approaches tend to yield models with lower predictive quality due to the relative weakness of the supervision provided by image-level labels. Towards reducing the performance gap between weakly and fully supervised training approaches, we propose to inject prior knowledge about object categories such as attributes (i.e., has a nose, is metal) into the model. Our primary motivation is to (i) increase knowledge sharing across object classes, and (ii) guide the concepts to be learned through the prior information available about the classes.

To inject prior knowledge into weakly-supervised approaches, we integrate embedding-driven prediction models, similar to labeled embedding techniques in zero-shot learning, into detection and segmentation models. We then train the embedding-based models via existing weakly-supervised training schemes. To evaluate this approach, we integrate it into the WSSDN [1] and PCL [7] models for weakly-supervised object detection and Puzzle-CAM [9] model for weakly-supervised semantic segmentation.

The experimental results on PASCAL VOC 2007 [4] and Microsoft COCO [3] datasets

for object detection and PASCAL VOC 2012 [8] for semantic segmentation suggests that the proposed methods for utilizing prior knowledge information can boost the object detection and semantic segmentation performance, especially on some of the object categories.

In Chapter 2 of this thesis, we present an overview the related work on object detection (2.1) and semantic segmentation (2.2) in both fully-supervised and weakly-supervised settings. In Chapter 3, we explain the datasets and metrics that are used in this thesis for both object detection and semantic segmentation. In Chapter 4, we explain the attributes and word embeddings as prior knowledge that are used in the experiments. In Chapter 5, our approach is explained and compared with baseline methods for weakly-supervised object detection. In Chapter 6, our approach is explained and compared with baseline methods for weakly-supervised semantic segmentation. In Chapter 7, our experiments on weakly-supervised object detection 7.1.1 and weakly-supervised semantic segmentation 7.2.1 are examined for Pascal VOC [4] and COCO 2014 [3] datasets. In the last Chapter 8, we conclude our thesis.

CHAPTER 2

LITERATURE REVIEW

This chapter focuses on the related works on object detection and semantic segmentation. We first briefly discuss object detection and semantic segmentation methods in Section 2.1 and Section 2.2, respectively. Both methods are discussed in detail, from Fully-supervised Methods (FS) to Weakly-supervised (WS) ones.

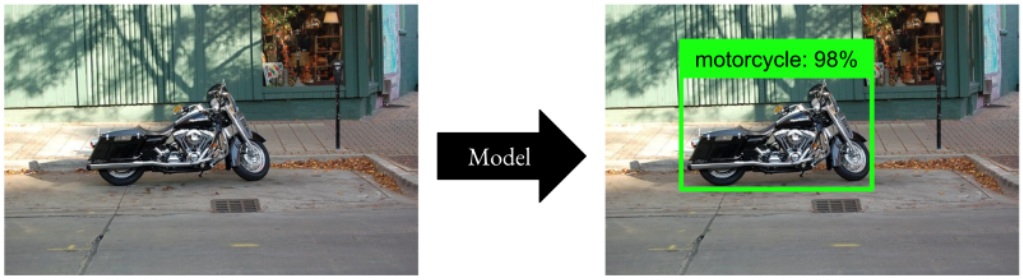
2.1 Object Detection

In this section, we review the literature for fully-supervised (Section 2.1.1) and weakly-supervised (Section 2.1.2) object detection.

Object detection uses computer vision models that locate and recognize objects in a video or image. Object detection uses bounding boxes to recognize objects and locate them in images or videos. An example of object detection inference is shown in Figure 2.1.

2.1.1 Fully-Supervised Object Detection

Recent studies have made great strides in object detection over the last few years. Commonly, these works have been in two ways. One area of research has been on proposal creation methods, intending to efficiently and accurately localize candidate objects. These methods are designed to produce high detection recall using fewer proposals as possible. The second area of research has been the study of training methods to classify proposals. Each proposal can be given a label that identifies it's background or foreground.



(a) input (b) output

Figure 2.1: An object detection inference example. Input (a) is given to the object detection model. The model predicts bounding box coordinates (x, y, width, height) as the green box, class category (motorcycle), and confidence score (98%) as output (b). The model can be both fully-supervised and weakly-supervised.

Images can contain objects in a variety of sizes and locations. Sliding windows used to be the most popular method for object detection shown in Figure 2.2. This is a comprehensive search process of an object for all window sizes and locations in an image. There are essential hyper-parameters to tune for sliding window approaches such as window size and stride count.

The object proposals reduce the number and computational cost of object detection. The algorithms have been shown to speed up object detection and improve it's performance significantly. The object proposal-based approaches can be beneficial in object detection [39], which can be seen as a preprocessing technique that improves computational efficiency and accuracy. Uijlings *et al.* [51] propose object proposals by hierarchically grouping superpixels at low levels. With a recall rate of 98% on Pascal VOC [2], and 92% on ImageNet [33], the number of proposals can reach around 2000. This significantly reduces the number of evaluated windows while ensuring a high detection rate.

Modern detectors are usually made up of two components: a backbone trained on ImageNet and a head used for predicting classes and bounding boxes shown in Figure 2.3. Examples for detection backbones include VGG [52], ResNet [53], ResNeXt [54].

The detection scheme can be designed as a one-stage or a two-stage approach. R-

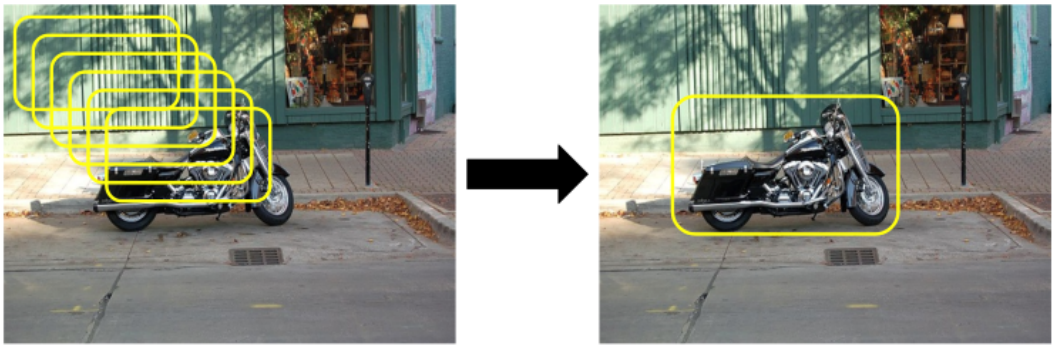


Figure 2.2: Sliding windows approach: A sliding window, in computer vision, is a rectangular area of fixed width and length that slides across an image. An image classifier is used to identify if each of these windows contains an object. This is a computationally intensive approach for object detection.

CNN [55] family contains perhaps the most well-known two-stage detectors. It includes Fast R-CNN [23], Faster R-CNN [24], R-FCN [25], Cascade R-CNN [26], Mask R-CNN [27] and Libra R–CNN [28]. RetinaNet [22], SSD [56], EfficientDet [57] and YOLO [17, 18, 19, 20] are among the best-known models for a one-stage detector. Inference process using these detectors are typically faster than two-stage detectors mainly thanks to skipping the proposal generation step, however, they also tend to perform relatively worse. To this end, more recently, anchor-free formulations has attracted interest in one-stage detection. For example, RepPoints [58] is an example of an anchor-free two-stage detector and CenterNet [59], CornerNet [60], DAFNe [61], and FCOS [21] are among the best-known models for an anchor-free one-stage detector. Noticeably, anchor optimization is not necessary with such anchor-free detectors.

Both one-stage and two-stage detectors have started to achieve impressive performance scores on Pascal VOC [2] and COCO [3] benchmarks. However, as they rely on fully-supervised training, all training images need to be manually annotated with the object bounding boxes, which is typically laborious, time-consuming, and can even be subjective and/or biased. WSL approaches, discussed in the next section, aim

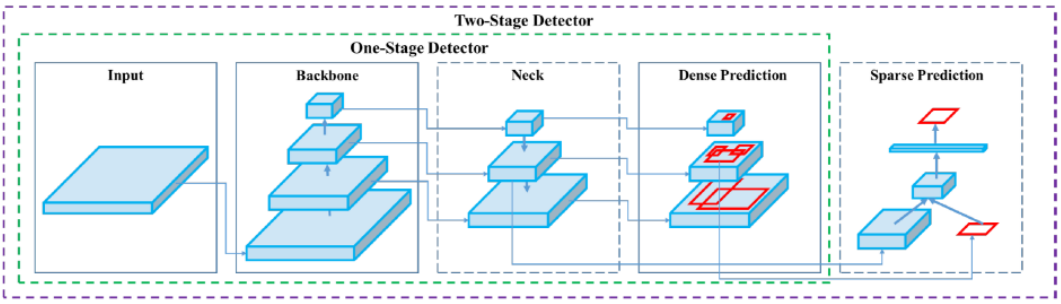


Figure 2.3: Object detection architecture overview. A one-stage detector includes input, backbone and neck, and dense prediction. A two-stage detector includes sparse prediction in addition to the one-stage detector. The head of a network is made up of dense and sparse predictions. Image is taken from [20].

to avoid these problems.

2.1.2 Weakly-Supervised Object Detection

Weakly-supervised object detection (WSOD) is an emerging topic the collection of large-scale data with image-level labels is much more easier than that with bounding box annotations, [37, 62, 41, 63, 64, 65, 66, 67, 36, 39, 40]. Chum *et al.* [63] initialize object locations using visual words. Then they present an exemplar model to determine the similarity between image pairs to update locations. Lazebnik *et al.* [65] use Felzenszwalb *et al.* [68] discriminatively trained part-based model to train a weakly-supervised object detection model. Deselaers *et al.* [64] use Alexe *et al.* [18] objectness approach to initialize proposals. Shi *et al.* [66] use Bayesian Joint modeling for object categories. Song *et al.* [67] aims to leverage discriminative regions to identify positive object windows with a latent SVM. Wang *et al.* [41] iteratively train detectors with relaxed multiple instance SVM (RMI-SVM) using derivable loss function. Several other works apply end-to-end approaches for WSOD [69, 70, 71, 7, 1, 10, 38].

Most of the existing WSOD approaches rely on a object proposal (also called candidate window) generation algorithm, such as Selective Search [51] or Edge Box [72] as the starting point. The goal is to find true positive instances and train an object proposal classifier. If an image is positive, there must exist at least one instance of

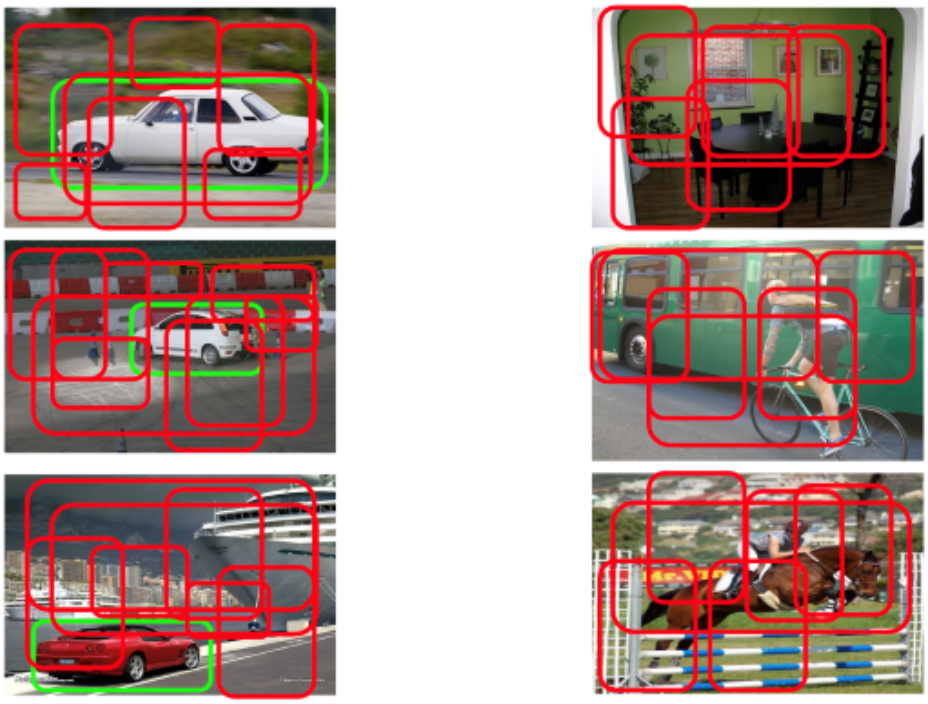


Figure 2.4: MIL (Multiple Instance Learning) approach. The images and candidate windows are interpreted as bags and instances, respectively. The goal is to find true positive instances and then train a window classifier. The "Car" classifier is shown in the example. Because images include a "car" instance, (a) can be considered positive bags. (b) is considered negative bags because the images do not contain a car instance.

the class is present by definition. If an image is negative, no instance of the class is present. The bounding box features are expected to be similar if there are objects of the same category in two different images. Based on this principle, WSOD methods aim to select boxes that have (discriminatively) consistent representations. These principles naturally lead to Multiple Instance Learning (MIL) based formulations [34, 35, 36, 1, 37, 38, 39, 40, 41].

In the MIL Framework, images and object proposals are interpreted as bags and instances. Each image is interpreted as a bag of regions, and supervision information specifies whether at least one example of a class is presented in the image or not, shown in Figure 2.4.

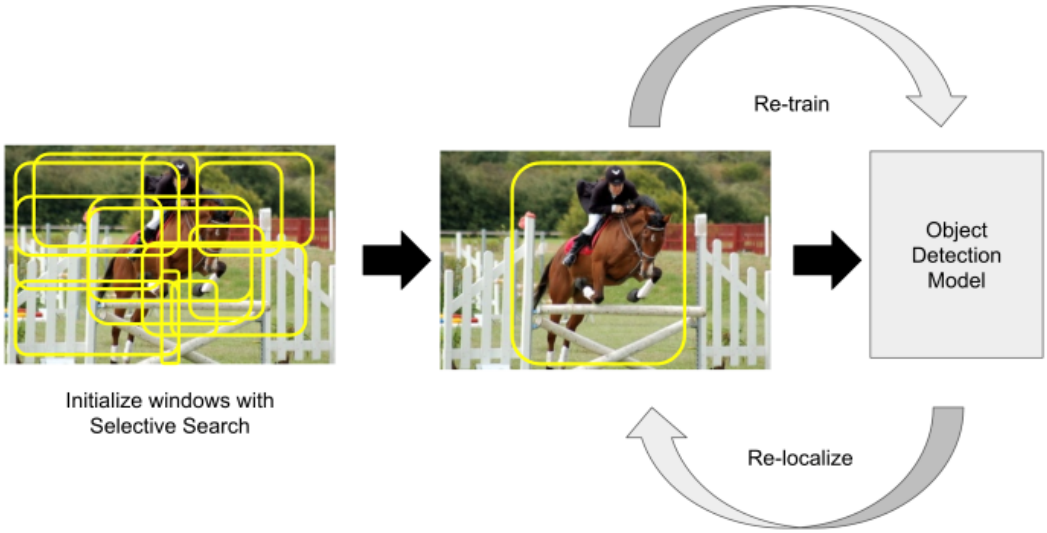


Figure 2.5: Standard MIL pipeline. First, candidate windows are generated with selective search [51] or edge-box [72] algorithms from the image. Then iteratively feed object detection model with candidate windows by re-training and re-localizing.

The goal of the MIL-WSOD is to find true positive instances and train the window classifier. Standard MIL pipeline is shown in 2.5, which typically contains the following stages:

- Initialize with candidate windows
- In a loop:
  - Re-train object detection model
  - Re-localize objects

While MIL is a widely used framework for developing WSOD approaches, it also has shortcomings. Two common problems are: (i) MIL training is highly dependent on initialization and can get stuck in local optima, and (ii) MIL-based WSOD models can unintentionally focus on only the most discriminative features, too, e.g., the head regions instead of the entire bodies of cats. Even if the model finds the head of a cat as a cat category, the detection is typically evaluated as a false positive due to missing out on the remainder of the ground-truth. To reduce the local minimum problem, Cinbis *et al.* [73] propose a MIL approach, based on Fisher vector image representation [74]

for detection windows, where a multi-fold training policy aims to avoid converging to poor local optima. Other approaches aim to MIL-based WSOD through a variety of other techniques [75, 76, 77, 78].

Another approach that uses information from well-trained image classification models is called Weakly-Supervised Deep Detection Network (WSDDN) [1] proposed by Bilen *et al.* WSDDN is the first end-to-end WSOD method and fine-tunes a pre-trained deep network. Their main idea is to exploit the knowledge accumulated over the CNN layers. WSDDN architecture feeds the input image into a pre-trained CNN-based classification model and replaces the last pooling layer with spatial pyramid pooling layer (region of interest pooling proposed in Fast R-CNN [23] is used in the experiments) so that the model becomes image size invariant. The output of the spatial pyramid pooling contains feature representation for each region proposal generated by Selective Search Windows (SSW) [51] or Edge Boxes (EB) [72] algorithm. The feature representations are then split into detection and recognition data streams. Finally, the two branches are combined and summed for each region to compute binary-log-loss. WSDDN architecture is shown in Figure 2.6. The overall simplicity of the formulation is one of the main advantages of WSDDN.

Singh *et al.* [79] investigates how common-sense information can learn detectors for new target categories. DOCK (Detecting Objects by transferring Common-sense Knowledge) [79] combines class similarity, attribute, spatial, and scene common-sense with WSDDN [1] architecture for region-level learning. Therefore, if a target class is not visually similar to any of the source classes, it's detector can be learned using other common-sense knowledge. The proposed method also adopts WSDDN [1] as a base detection network, but the method uses bounding-box annotations on source categories and image-level annotations on target categories for training.

One of the major problems in WSOD (Weakly-Supervised Object Detection) is that the model focuses on the discriminative parts of the objects rather than the whole object extent. To solve this problem, Tang *et al.* [10] proposes OICR (Online Instance Classifier Refinement) algorithm. The top-scoring proposal only contains discriminative parts of the object, but the proposals that have high overlap with the top-scoring proposal contain larger parts of the objects. Almost 2000 object proposals are gen-

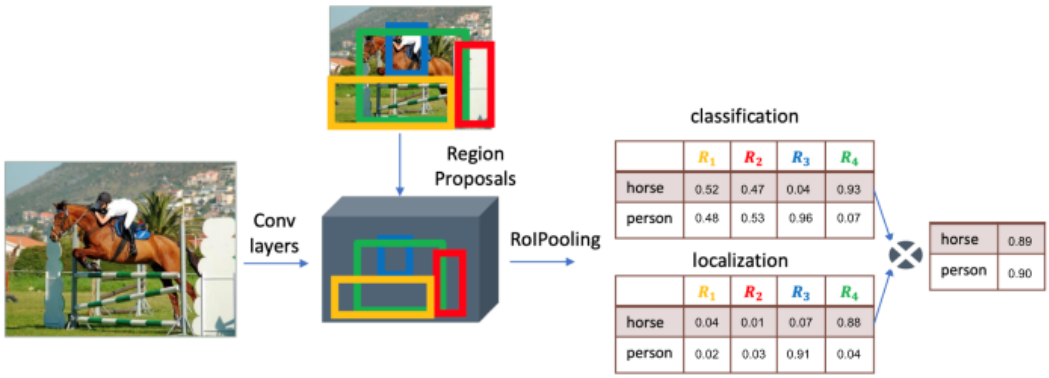


Figure 2.6: WSDDN (Weakly-Supervised Deep Detection Network) [1] pipeline. Image and candidate windows generated with SSW [51] are fed into ROI Pooling and then divided into two-stream architectures. The classification stream assigns each region to a class. The localization region picks the most promising windows in an image given a class. The figure is taken from Hakan Bilen's CVPR'18 presentation.

erated with the Selective Search [51] method. Proposal scores are generated after applying WSDDN [1]. These proposals and scores are used as supervision for the first refinement branch, shown in Figure 2.7.

Different proposals address different parts of the objects. Although all these proposals could be considered "cat", only those with sufficient ground truth and IoU can make correct detections, as shown in Figure 2.8.

Some top-ranking candidate boxes may be limited to locating parts of objects rather than whole objects. The detection of objects requires that the boxes resulting from these classifications correctly classify and localize them and have sufficient overlap with ground truth boxes.

Proposal clusters can be created by treating ground truth bounding boxes and their centers as cluster centers for fully-supervised object detection. The object detectors can then be trained according to the proposed clusters. This solves the problem of detectors focusing on only parts. It is impossible to generate proposal centers in

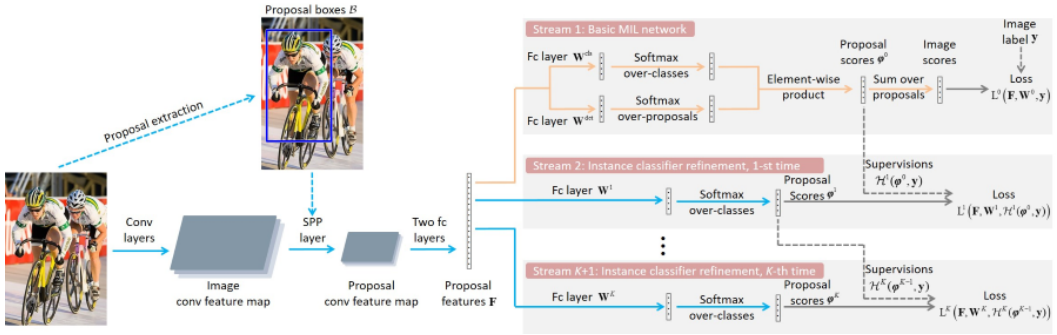


Figure 2.7: OICR [10] architecture. Candidate proposals are generated with SSW [51] and their scores are used as supervision for the next refinement stream. After some refinements, the network will not discriminate only parts of the objects but the whole extent. The figure is taken from OICR [10].

weakly-supervised object detection because ground truth bounding boxes are unavailable. Because these top-ranking proposals can always detect parts of objects, Tang *et al.* [7] with PCL (Proposal Cluster Learning) propose to create proposal cluster centers from proposals with high classification scores during training. This means that first select proposals with high proposal scores for each image to become cluster centers. Next, the proposal clusters will be generated based upon spatial overlaps with these cluster centers. Tang *et al.* [7] propose a graph-based approach to find the cluster centers. Figure 2.9 (a) represents the traditional MIL (Multiple Instance Learning) approaches, (b) represents the proposal label assignment, and (c) represents the proposal cluster as a bag. “0”, “1”, and “2” denotes the classes of the objects.

Kantarov *et al.* [38] proposes context-aware models for WSOD by introducing additive and contrastive models that use neighbor regions to increase detection performance by reducing the focus on discriminator regions.

There are some challenges in WSOD, including instance ambiguity, part domination, and memory consumption. Ren *et al.* [70] proposed instance-aware, context-focused, and memory-efficient WSOD methods to solve these challenges. The paper introduces multiple instance self-training (MIST), a teacher-student distillation process to solve instance ambiguity. The paper also proposes Concrete Dropblock to focus on context rather than object parts to solve part domination. With concrete dropblock, discriminative object parts are discarded. They proposed Sequential back-

Figure 2.8: Different candidate windows for "cat" category. These windows can all be considered as "cat" classes, but only the one with the green box that indicates IoU is sufficient is considered as "cat" for the network.

propagation to solve the high memory consumption on training a network.

The performance of weakly-supervised object detection methods has increased noticeably in recent times. But the gap between fully-supervised and weakly-supervised object detection is still very significant as fully-supervised detections performances have also been increasing, e.g., on both the COCO test dataset [80, 81, 82, 83, 84] and the Pascal VOC 2007 validation dataset [85, 86, 87, 88, 89].

Finally, we note that there are other weakly supervised detection approaches that do not strictly fit into the standard WSOD setting. For example, a well-known idea based on transfer-learning is learning a mapping between image classification and object detection tasks and then using this mapping to generalize object detection on instances of the test classes. Hoffman *et al.* proposed Large Scale Detection through Adaptation (LSDA) [90], in which they split classes into source classes with instance-level labels and target classes with only image-level labels. They trained a generic classifier to detector transformation network by using the correlation between similar source and target categories. Related to our work on this different problem, Tang *et*

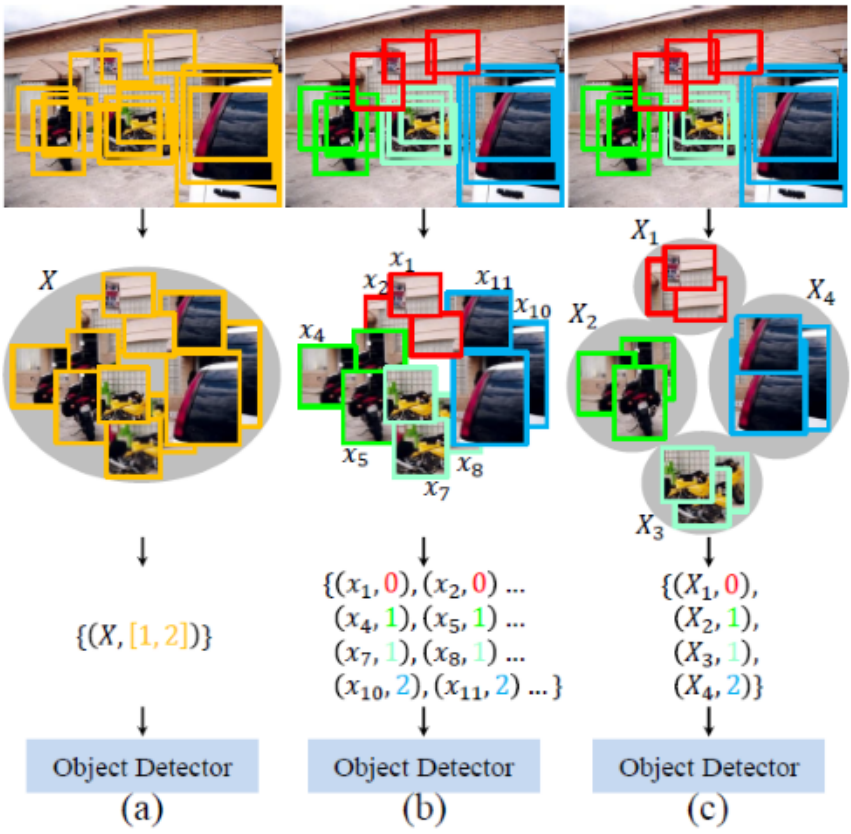


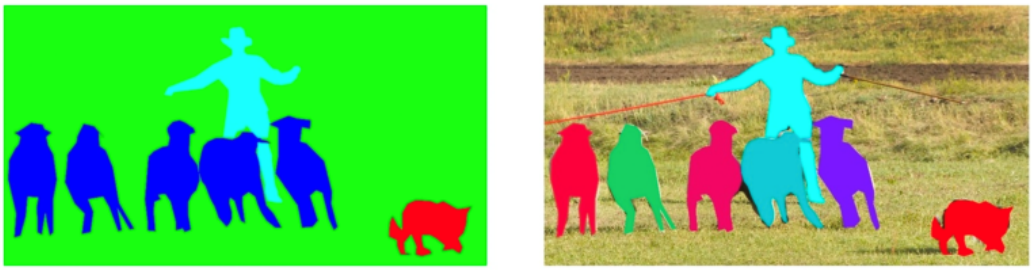
Figure 2.9: Comparison of PCL [7] and traditional approaches. (a) is the traditional MIL (Multiple Instance Learning), (b) the proposal label assignment and (c) the proposal cluster as a bag. The categories of the objects are indicated by the numbers "0", "1", and "2". The figure is taken from PCL [7].

al. [91] enhances LSDA by injecting visual and semantic domain knowledge during the classifier-to-detector process.

2.2 Semantic Segmentation

In this section, we present an overview of the literature for fully-supervised (2.2.1) and weakly-supervised (2.2.2) semantic segmentation.

Image segmentation plays a crucial role in computer vision tasks. There are many applications for image segmentation areas, including medical imaging (tumor segmentation, pneumonia parts, cell segmentation), autonomous driving (segmenting traf-



(a) semantic segmentation (b) instance segmentation

Figure 2.10: Example outputs of semantic (a) and instance (b) segmentation. Turquoise, blue and red colors represent person, sheep and dog respectively in (a). In semantic segmentation, there is no distinction on the same category objects. All sheep colors are blue in (a). In instance segmentation, all objects with the same category are handled separately. All sheep colors are different in (b). MSCOCO dataset [3] contains instances segmentation annotations. The figure is taken from MSCOCO [3].

fic lights, cars, pedestrians), augmented reality (segmenting parts of objects), video surveillance, geo-sensing. With the improvement of deep learning algorithms, success in segmentation has also increased.

We can formulate the image segmentation models as per-pixel contextual classification models. We should also note the important difference between semantic and instance segmentation methods. In semantic segmentation, the aim is to classify every pixel with a class category such as a person, cat, dog. It treats objects in the same category as a single object. In instance segmentation, however, the aim is to classify every pixel with a class category and detect the objects separately. There is a distinction between objects of the same category. An example of semantic and instance segmentation is shown in Figure 2.10. The ground truth of images for fully-supervised image segmentation methods is class categories for every pixel.

In the following sections, we first go over fully-supervised semantic segmentation methods in Section 2.2.1 and then weakly-supervised semantic segmentation methods in Section 2.2.2.

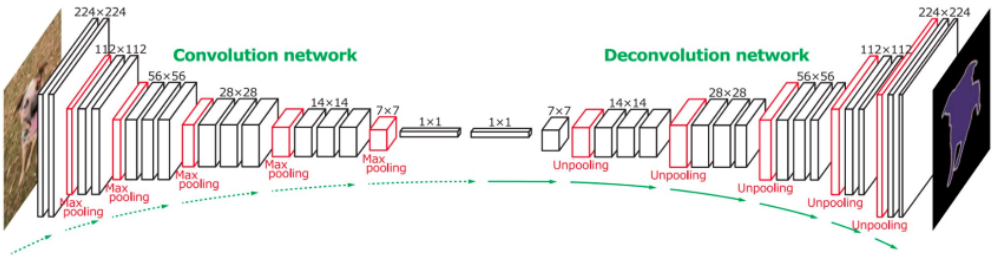


Figure 2.11: DeConvNet [11] architecture. The left side of the network uses VGG-16 [52]. The right side of the network uses deconvolution and unpooling to generate a segmentation map of the image. The figure is taken from DeConvNet [11].

2.2.1 Fully-Supervised Semantic Segmentation

Fully-supervised semantic segmentation requires class categories for every pixel as ground truth. Variety of methods try to solve this problem including traditional methods such as k-means and subtractive clustering [92], conditional random fields [93], statistical region merge [94] and deep learning methods such as UNet [12], DeepLabv3 [95], atrous convolution [96], PSPN (Pyramid Scene Parsing Networks) [97].

Fully Convolutional Networks (FCN) [98] is trained end-to-end with arbitrary image sizes, as proposed by Long *et al.* This network includes only convolutional layers, so that image size can be different. Fully-connected layers are removed from CNN architectures such as AlexNet [99] and VGGNet [52] to support arbitrary sized images. For the limitations of FCN, it's not suitable for real-time applications because it's very compute expensive and tends to focus on parts of the objects rather than global context. To solve the global context problem, Liu *et al.* proposed ParseNet [100] by extracting global feature vector with global pooling of feature map.

Most of the recent research on semantic segmentation use encoder-decoder based algorithms. Noh *et al.* proposes DeConvNet [11] by introducing deconvolution operation. The first part of the network uses VGG-16 [52] as convolution layers, the later part of the network uses deconvolution operation to match the image size. Convolution and deconvolution operations in DeConvNet are illustrated in Figure 2.11.

SegNet [29] is a fully convolutional encoder-decoder based network, proposed by Badrinarayanan *et al.* SegNet [29] is similar to DeConvNet [11] but SegNet uses nonlinear up-sampling method on decoder part of the network.

Yuan *et al.* proposes HRNet [30] to solve the loss of image information on the encoder part by keeping the high-resolution part of the features and feeding them to the last layers.

Ronneberger *et al.* proposes UNet [12] that uses fully convolutional endoder-decoder architecture for biomedical image segmentation. UNet contains contracting (encoder) and expansive (decoder) paths. The contracting path uses typical convolutional and max-pooling layers to grab the context in the input image. The expansive path uses deconvolution layers to get a precise output segmentation map, which is symmetric to the contracting path. Input image of UNet can be any size, as it is FCN (Fully Convolutional Network) and does not contain dense layers. UNet network is shown in Figure 2.12.

A variety of applications that uses UNet [12] architecture on different image domains have ben published. Cicek *et al.* proposes 3D-UNet [31] for 3D images, Zhang *et al.* [101] proposes road extraction based on UNet. Zhou *et al.* proposes UNet++ [102] to increase the validation scores of UNet by redesigning skip pathways. VNet is an another FCN based 3D image segmentation model, proposed by Milletari *et al.* [103].

Multi-scale and pyramid networks are commonly used in semantic segmentation. Lin *et al.* proposed FPN (Feature Pyramid Networks) [13] which is mainly used by object detection tasks but can be used on semantic segmentation as well. FPN architecture for semantic segmentation is shown in Figure 2.13.

Learning the global context is a problem in semantic segmentation. Zhao *et al.* proposed PSPN (Pyramid Scene Parsing Networks) [97], which learns the global context by concatenating the initial feature layers with the last layers on different pyramids. To get the semantic maps convolution layer is used. There are other semantic segmentation methods that uses pyramid and multi-scale fashion such as APC-Net [104], MSCI [105], salient segmentation [106].

Dilated (atrous) convolution is used to shrink the size of features by adding space

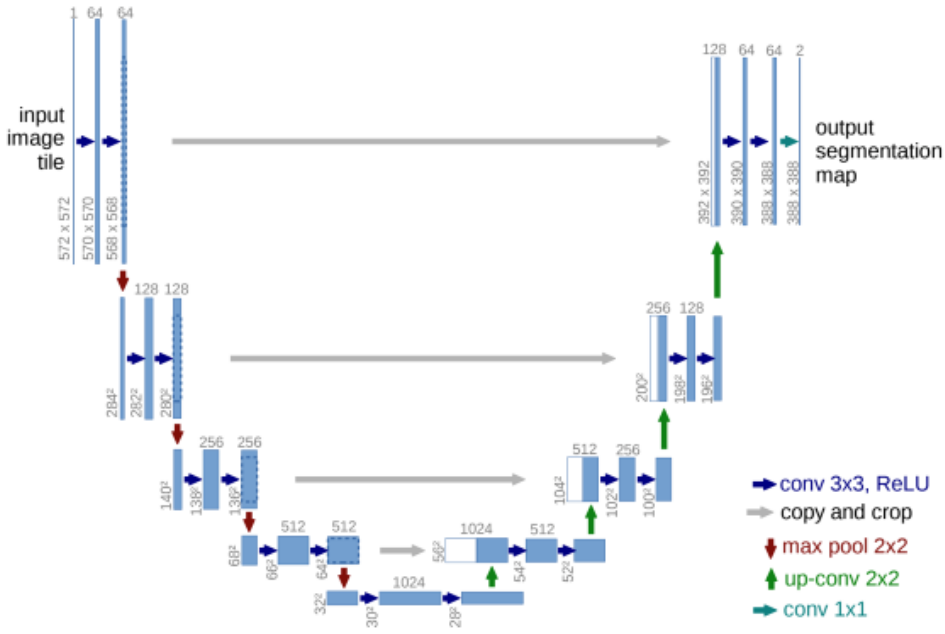


Figure 2.12: UNet [12] architecture. The left side of the network is called contracting (encoder) path. The right side of the network is called expansive (decoder) path. The last layer of the expansive part is 1×1 convolution. The figure is taken from UNet [12].

between the weights. Some works on real-time semantic segmentation use dilated convolutions. Chen *et al.* proposes DeepLabv1 [107] and DeepLabv2 [14] which segments images in multiple scales using (ASPP) Atrous Spatial Pyramid Pooling and decrease the feature size by dilated convolution layers. DeepLabv2 [14] architecture is shown in Figure 2.14. DeepLabv3 [95] was proposed by Chen *et al.* to increase the validation score of semantic segmentation by introducing batch normalization and 1×1 convolution for (ASPP) Atrous Spatial Pyramid Pooling. Chen *et al.* proposes DeepLabv3++ [108], which removes batch normalization and max pooling by using Xception [109] (depthwise separable convolutions) as backbone.

Some other works propose to handle semantic segmentation problem by using RNNs (Recurrent Neural Network), such as ReSeg [110], ReNet [111], Graph LSTM [112]. Attention based and GAN models are on the rise for semantic segmentation problem including Attention to Scale [112], reverse attention [113], pyramid attention

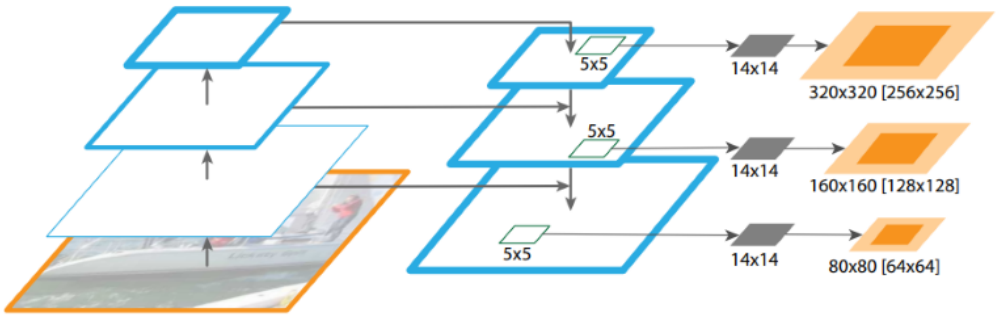


Figure 2.13: FPN (Feature Pyramid Networks) for semantic segmentation [13] architecture. MLP (Multi-Layer Perceptron) is applied to obtain the semantic masks. The figure is taken from FPN [13].

[114], GAN methods [115, 116, 117]. Most state-of-the-art methods, however, use the DeepLabv3++ [108] feed-forward network in their models.

2.2.2 Weakly-Supervised Semantic Segmentation

Fully-supervised semantic segmentation methods have made remarkable progress in recent years. Weakly-supervised semantic segmentation methods aim to close the performance gap with fully-supervised semantic segmentation methods. Weak labels such as bounding boxes, scribbles, points, and image-level labels are used on weakly-supervised semantic segmentation methods. In this thesis, we focus on weakly-supervised semantic segmentation using image-level labels as supervision.

A major challenge in weakly-supervised semantic segmentation is the tendency of the network to focus on discriminative parts such as the head of a bird instead of the whole part. For image-label-based weakly supervised segmentation, most of the state-of-the-art methods rely on CAM (Class Activation Map) [42] for pseudo-label creation used as ground truth semantic masks, such as [43, 15, 44, 45, 46, 47]. However, CAM-based approaches also suffer from focusing on most discriminative regions only.

Class activation maps (CAM) may not be consistent across different scales of input.

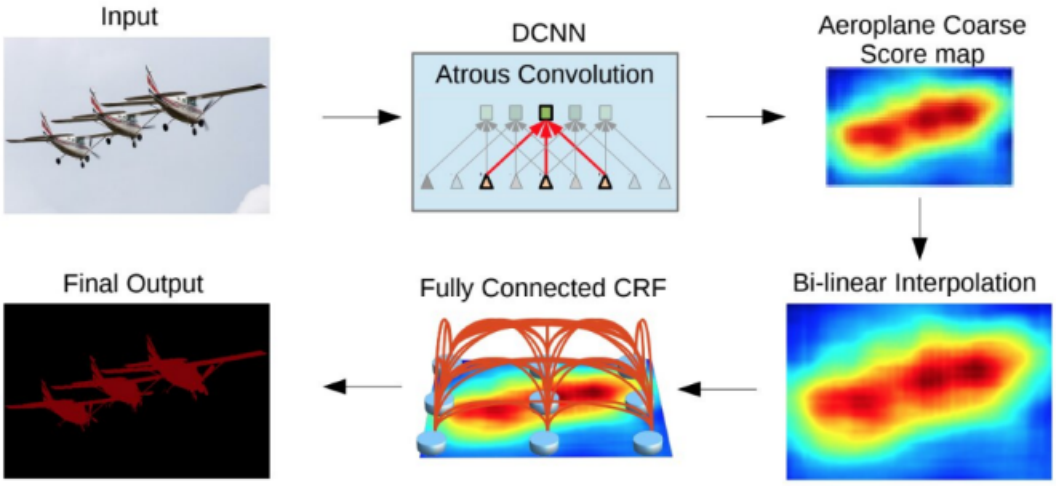


Figure 2.14: DeepLabv2 [14] architecture. Atrous convolution is used for generating a class score map. After bi-linear interpolation of the score map, fully connected CRF (Conditional Random Field) is used to get the final predictions. The figure is taken from DeepLabv2 [14].

To address this problem, Wang *et al.* propose SEAM [15] which reduces the difference between the different scaled inputs using a siamese network. The architecture of SEAM [15] is shown in Figure 2.15.

Wei *et al.* [118] proposes Adversarial Erasing that mines discriminative regions and erase them (head of an object) progressively over and over until the discovery of the object's whole body. This method is similar to concrete DropBlock, which drops discriminative parts features, as proposed by [70] for the object detection problem. This method, however, is a slow approach because the network needs to be trained repeatedly to produce dense localization maps.

Similar to the fully supervised approaches, dilated (atrous) convolution is also used by weakly-supervised semantic segmentation techniques. For example, Wei *et al.* [119] proposed dilated convolutions to reduce the focus on discriminative regions and obtain dense segmentation maps by different dilation rates. This approach does not require repetitive training. The final activation map is obtained by a weight com-

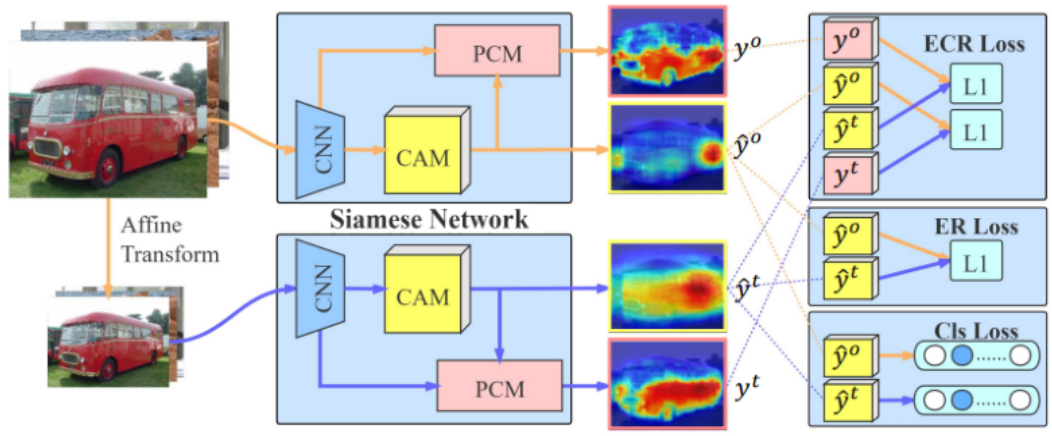


Figure 2.15: SEAM [15] architecture. Multi-scaled input images are fed into the classification network and using loss for consistency between different scales. The figure is taken from SEAM [15].

bination of activation maps of different dilation rates. The last segmentation mask is generated from the final activation map and used for ground truth of the network. This approach can be also used for semi-supervised learning.

Another approach in weakly-supervised semantic segmentation is to apply adversarial attacks [120] to get the whole target object instead of parts of the target object. Lee *et al.* [121] proposes to get manipulated image by finding a perturbation of the input image to increase the classification score by move away from decision boundary. The manipulation of image is discovered by the adversarial climbing approach. This approach can be applied on weakly, and semi-supervised semantic segmentation and does not require re-training the models.

The initial CAMs of weakly-supervised semantic segmentation networks may not cover the whole target objects. Jo *et al.* proposes Puzzle-CAM [9] which divides the image into tiles and generates CAMs from these tiles. The difference between full input and tiled images is reduced with reconstruction loss. The architecture of Puzzle-CAM is shown in Figure 2.16.

We use the state-of-the-art approach Puzzle-CAM [9] as our baseline in the weakly-supervised semantic segmentation experiments in this thesis.

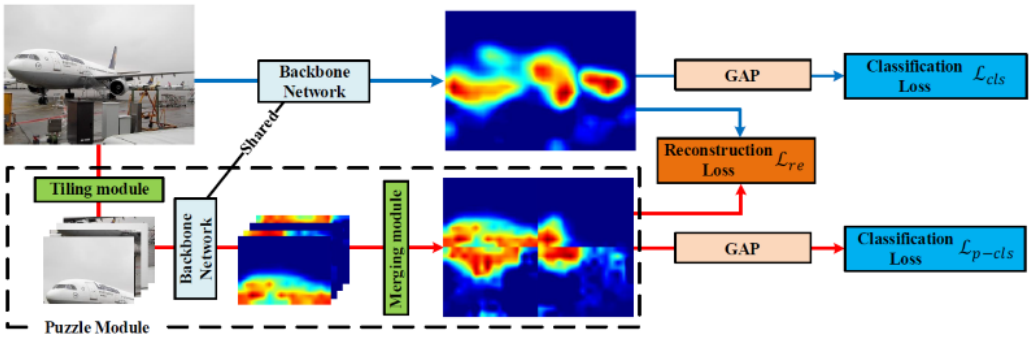


Figure 2.16: Puzzle-CAM [9] architecture. The input image is divided into tiled parts. CAMs are generated from tiled parts and then merged into a single CAM using the merging module. Another CAM is also generated using the original input image. The difference between full input and tiled images is reduced with reconstruction loss. The figure is taken from Puzzle-CAM [9].

CHAPTER 3

DATASETS AND METRICS

3.1 Datasets

Datasets have been a crucial part of object detection research. They serve as a critical component of performance evaluation in comparing competing approaches and push the research to address progressively more challenging problems. Access to large image collections online makes it possible to create extensive datasets with different categories. Large-scale datasets with millions of images have opened the door to significant breakthroughs that have enabled object recognition at exceptional speeds.

In order to collect large amounts of data sets, first of all, the object categories must be determined. Once the categories have been determined, it is necessary to collect large quantities of videos or photos for each category. Open-source data on the internet can be used when collecting videos and photos. If the data set needs to be collected for a specific area, video capture may be required.

Once the photos or videos are collected, annotating is required for the determined categories. Distributed human labor, such as Amazon Mechanical Turk, is generally used for long and tedious labeling operations. Labeling time varies according to the labeling type (image-level, bounding-box, pixel). After the labeling is completed, optionally, cross-check validation can be done across human annotators. Generally, these steps need to be completed to create a dataset.

Datasets of various sizes and domains have been collected. Mnist [122], Scenes15 [123, 124, 125], Tiny Images [126], ImageNet [33], MSCOCO [3], Pascal VOC [2, 4, 8], Sun [127], Caltech101 [128, 129], Caltech [130, 131], Open Images [132] are

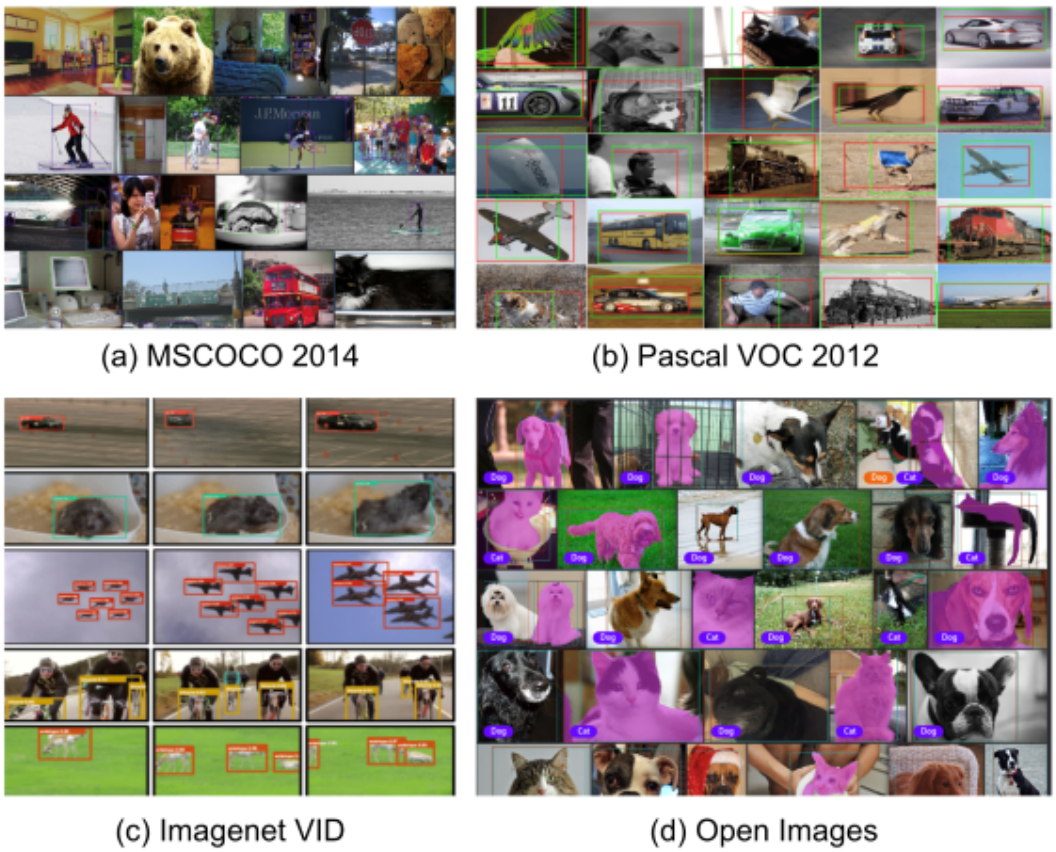


Figure 3.1: Example images from different computer vision datasets. (a) Example images from COCO 2014 dataset [3]. (b) Example images from Pascal VOC 2012 dataset [8]. (c) Example images from ImageNet VID dataset [33]. (d) Example images from Open images dataset [132]. The images in this figure are taken from the stated datasets.

just a few examples. Some images from different datasets are shown in Figure 3.1.

The most commonly used datasets in object detection are MSCOCO [3], Pascal VOC [2, 4, 8] and ImagenetVID. The statistics of these datasets are shown in Table 3.1.

The Pascal VOC dataset [4] collection started in 2005 with five categories and eventually was enlarged to 20 categories. The categories in the dataset include objects in daily life. The categories in the dataset are as follows: aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, diningtable, dog, horse, motorbike, person, pottedplant, sheep, sofa, train, tvmonitor. The Pascal VOC dataset [4] contains bounding box and pixel-level annotations for object detection and segmentation tasks.

Table 3.1: The statistics of the most commonly used datasets in object detection problem. We can understand how many objects are in an image from the Objects / Image column. As can be seen, there are more objects in the images in the MSCOCO dataset [3] than in the other datasets. The dataset with the highest category (200) is Imagenet VID dataset [33].

Dataset	**Total Images**	**Categories**	**Objects / Image**
Pascal VOC 2012 [8]	11540	20	2.4
MSCOCO 2014 [3]	328000	80	7.3
Imagenet VID [33]	116159	200	2.8

The collection of the Imagenet dataset had started in 2009. In 2015, it was announced as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [33]. Imagenet VID was primarily created to address the data insufficiency observed on Pascal VOC and its limited number (20) of categories. There are 200 categories in the Imagenet VID dataset. Some of the categories in the Imagenet VID dataset are as follows: accordion, airplane, ant, antelope, apple, armadillo, artichoke, axe, baby bed, backpack, bagel, balance beam, banana, band aid, banjo, baseball, basketball, bathing cap, beaker, can opener, car, cart, cattle, cello, centipede, computer keyboard, punching bag, purse, rabbit, racket, ray, red panda, refrigerator, remote control, rubber eras whale, wine bottle, zebra. The Imagenet dataset [33] contains bounding box annotations for object detection task.

Imagenet VID dataset [33] images usually contain 1 or 2 objects. These objects are often large and located in the center of the image. Models should be able to detect small objects in dense scenes, just as they would in real life. MSCOCO [3] was created to bring the dataset closer to reality. MSCOCO has since become the de-facto standard in object recognition problem.

The common class categories for the datasets of interest are shown in Figure 3.2.

In this thesis, we use Pascal VOC [4] and MSCOCO [3] datasets for the weakly-supervised object detection experiments, and Pascal VOC [8] dataset for the weakly-supervised semantic segmentation experiments.

3.2 Metrics

Various evaluation metrics are used for object detection and semantic segmentation. We summarize the mainstream object detection metrics in Section 3.2.1 and the semantic segmentation metrics in Section 3.2.2.

3.2.1 Object Detection Metrics

Many competitions including Pascal VOC [4], MSCOCO [3] push towards *solving* the object detection problem. Some competitions create object detection metrics for evaluation. The most used metrics for object detection tasks are AP (average precision) and mAP (mean average precision). There are some differences in AP calculations on COCO and VOC challenges. The main goal of the object detection task is to bring the detected objects closer to the ground truth objects.

A detection from a model typically consists of three elements: **confidence score** (between 0 and 1), **class category** and **bounding box** (x, y, w, h). The closeness between the predicted bounding box and ground truth bounding box can be measured by IoU (Intersection Over Union) based on Jaccard index [134], or similar metrics. Assuming that the ground truth bounding box is BB_{gt} and predicted bounding box is BB_p, the IoU of BB_{gt} and BB_p is intersection of BB_{gt} and BB_p divided by union of BB_{gt} and BB_p (Equation 3.1). IoU value is between 0 and 1, where 1 means a perfect match. If there are no overlaps between bounding boxes, IoU will be zero. An example of IoU between ground truth and predicted bounding boxes is shown in Figure 3.3.

$$IOU = \frac{area(BB_{gt} \cap BB_p)}{area(BB_{gt} \cup BB_p)} \tag{3.1}$$

To calculate AP (Average Precision) we need these terms:

- TP (True Positive): Prediction bounding box is correct. $IoU > Threshold$.
- FP (False Positive): Prediction bounding box is wrong. $IoU < Threshold$.

- FN (False Negative): Ground truth bounding box is not detected.
- TN (True Negative): Not considered for object detection domain. There are many bounding boxes that the model is not predicted and are not ground truth.

Precision is the proportion of true positive bounding boxes across all detected bounding boxes. The purpose of precision is to try to find how many of the detected bounding boxes are correctly classified. The precision formula is shown in Equation 3.2:

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{\text{All Detections}} \tag{3.2}$$

Recall is the proportion of true positive bounding boxes across all ground truth bounding boxes. The purpose of recall is to find all ground truth bounding boxes. The recall formula is shown in Equation 3.3:

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{\text{All Ground Truths}} \tag{3.3}$$

There is a trade-off between precision and recall. If the model predicts more bounding boxes (high recall), the predicted bounding boxes will be less correct (low precision). The precision value should not decrease too much when the recall increases to understand a good detector.

AP (Average Precision) metric is calculated with area under the precision-recall curve for each category. Recall is the x-axis and precision is the y-axis of the plot. An example precision-recall plot is shown in Figure 3.4. Mean average precision (mAP) is the average of all Average Precisions (AP) for each category.

Different challenges interpret the AP value differently. The AP values interpreted by Pascal VOC and COCO challenges are shown in the Table 3.2. In this thesis, we used AP with 0.5 IoU for Pascal VOC [4, 8] datasets and AP with .50:.05:.95 IoU for COCO dataset [3].

3.2.2 Semantic Segmentation Metrics

Semantic segmentation metrics are similar to object detection metrics. Instead of calculating metrics with bounding box areas, pixels are used on semantic segmentation

Table 3.2: Different AP (Average Precision) metrics for Pascal VOC [2] and COCO [3] challenges. AP with .50:.05:.95 IoU is the primary challenge metric for COCO. AP with .50 IoU is equal to mAP in COCO challenge.

Metric	**IoU**	**Challenge**	**Description**
AP	0.5	Pascal VOC [2]	AP is calculated with a fixed 0.5 IoU threshold for all recall points.
mAP	0.5	Pascal VOC [2]	Average of each classes AP values.
AP	.50:.05:.95	MSCOCO [3]	AP is calculated with 10 different IoU values [0.50, 0.55, 0.60, ..., 0.95] and taking the average of them.
AP	.50	MSCOCO [3]	AP is calculated with a fixed 0.50 IoU threshold for 101 recall points.
AP	.75	MSCOCO [3]	AP is calculated with a fixed 0.75 IoU threshold for 101 recall points.
AP^{small}	.50:.05:.95	MSCOCO [3]	AP is calculated for small objects whose area is less than 32^2 pixels.
AP^{medium}	.50:.05:.95	MSCOCO [3]	AP is calculated for medium objects whose area is between 32^2 and 96^2 pixels.
AP^{large}	.50:.05:.95	MSCOCO [3]	AP is calculated for large objects whose area is greater than 96^2 pixels.

validations.

Mean Intersection over union (mIoU) is the primarily used metric for semantic segmentation models. IoU equation for object detection is shown in Equation 3.1. Instead of using a bounding box for the IoU calculations, pixel-based TP, FN, FP are used for semantic segmentation. First of all, IoU is calculated for each class and then the average of IoUs is taken to get the mIoU values. IoU calculation for semantic segmentation is as follows:

$$Jaccard = IOU = \frac{TP}{TP + FP + FN}. \tag{3.4}$$

Dice coefficient [135] (F1) is first published by Sorensen *et al.* to measure the similarities of two samples. This metric is used to evaluate semantic segmentation models.

The Dice coefficient is equal to F1 metric. The formula of Dice is similar to IoU, shown in Equation 3.5:

$$F1 = Dice = \frac{2TP}{2TP + FP + FN} \tag{3.5}$$

The most commonly used metric in semantic segmentation is mIoU.

3.3 Summary

This chapter presents an overview of the datasets and methods required to evaluate object detection and semantic segmentation models. Fully-supervised object detection and semantic segmentation models are evaluated in terms of AP and mIoU. Weakly-supervised object detection and semantic segmentation models are mainly evaluated in the same way.

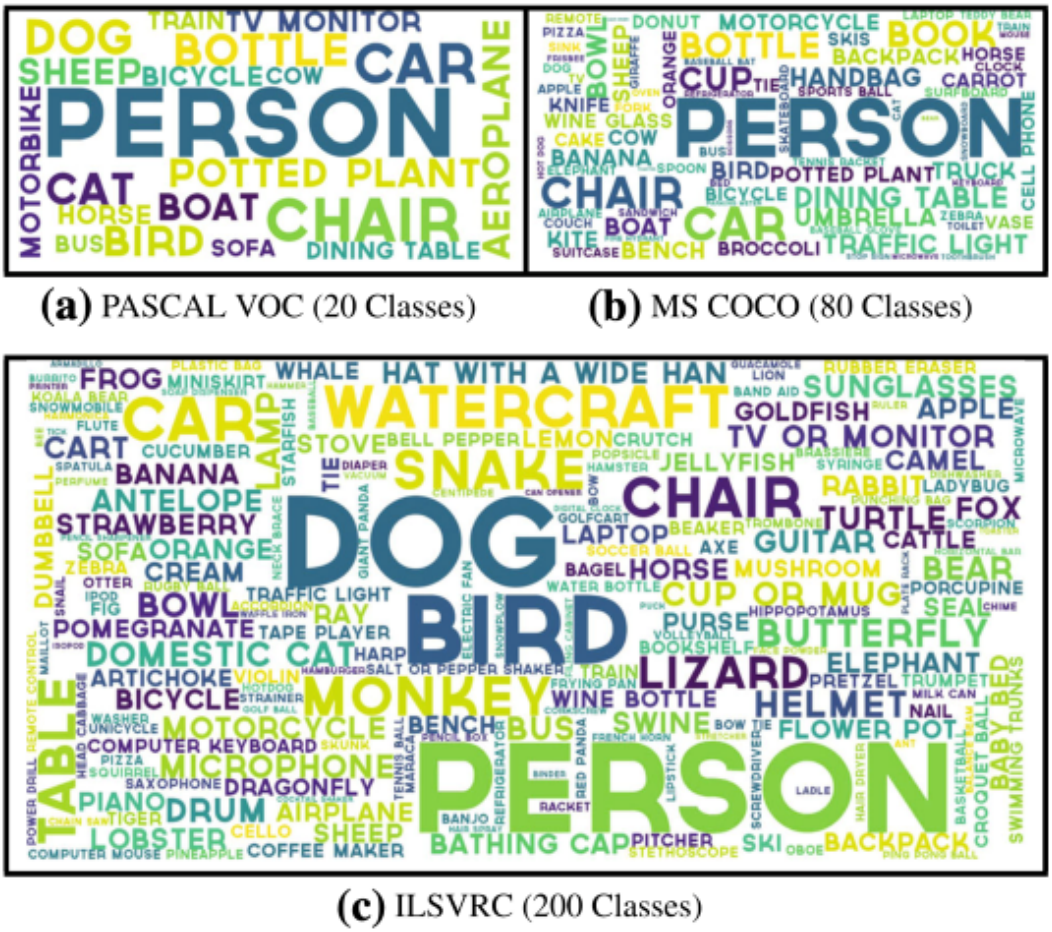


Figure 3.2: The most common class categories for different datasets. All datasets contains person category as the most. (a) Category frequency map for Pascal VOC dataset [4]. (b) Category frequency map for MSCOCO dataset [3]. (c) Category frequency map for Imagenet VID dataset [33]. The images in this figure are taken from Liu *et al.* [133].

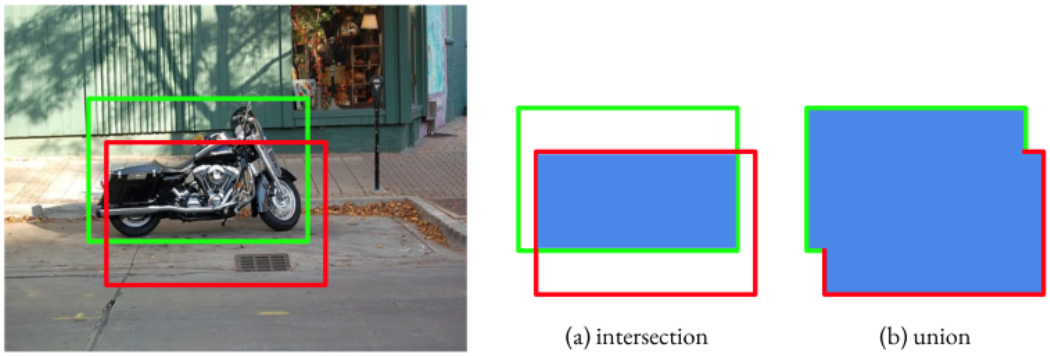


Figure 3.3: An IoU (intersection over union) example. The green and red bounding boxes indicate ground truth and prediction, respectively. (a) Intersection (overlap) of green and red bounding boxes. (b) Union of green and red bounding boxes.

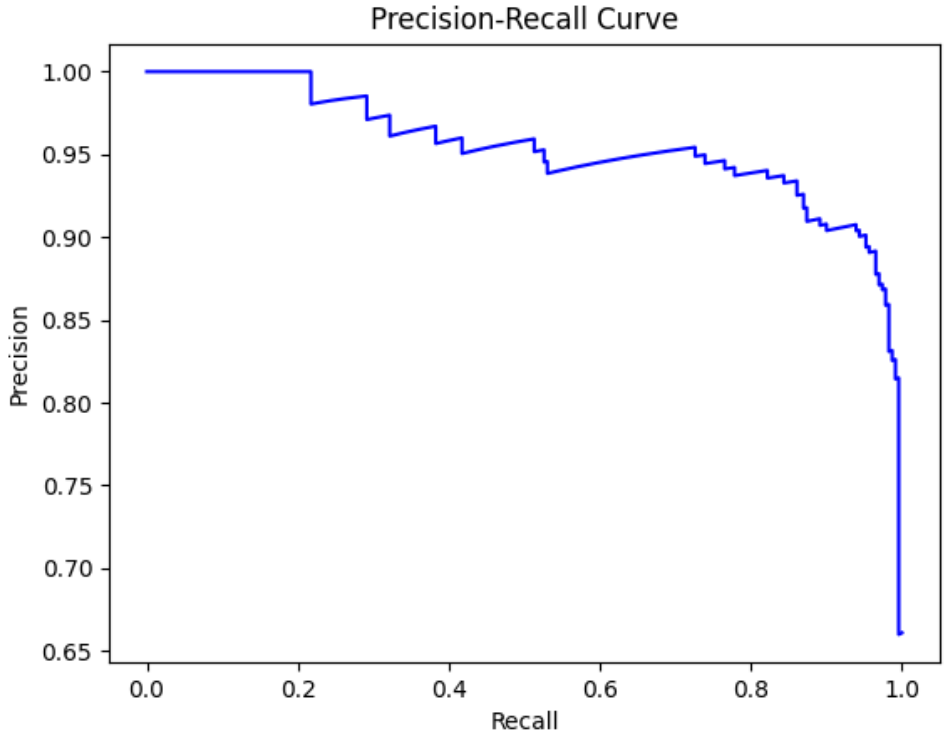


Figure 3.4: An example of Precision-Recall curve for one category. As recall increases, precision decreases. The area under the blue line is Average Precision (AP) for one category.

CHAPTER 4

PRIOR KNOWLEDGE

4.1 Overview

In this chapter, we provide an overview of the prior knowledge sources that we utilize for the purpose of improving weakly supervised training techniques. In particular, we focus on class attributes and class name word embeddings as the prior knowledge on classes of interest. In the sections following, we present their details for the PASCAL and MSCOCO datasets.

4.2 aPascal

Farhadi *et al.* [16] has introduced two attribute-based recognition datasets built on Pascal VOC dataset [4] and Yahoo Image search images. The attribute dataset collected with Pascal is called aPascal, and the dataset collected with Yahoo is called aYahoo. In this thesis, we work with aPascal attributes.

For each train and test image in Pascal VOC dataset [4], the images are annotated in terms of **64** attributes. The annotation process was carried out by Amazon's Mechanical Turk.

The attributes are categorized into three types:

- **Shape**: Form of an object such as Round, 3D Boxy, 2D Boxy.
- **Material**: What material is it made of such as Metal, Plastic, Wool.
- **Part**: Visible parts of an object such as Arm, Leg, Face.

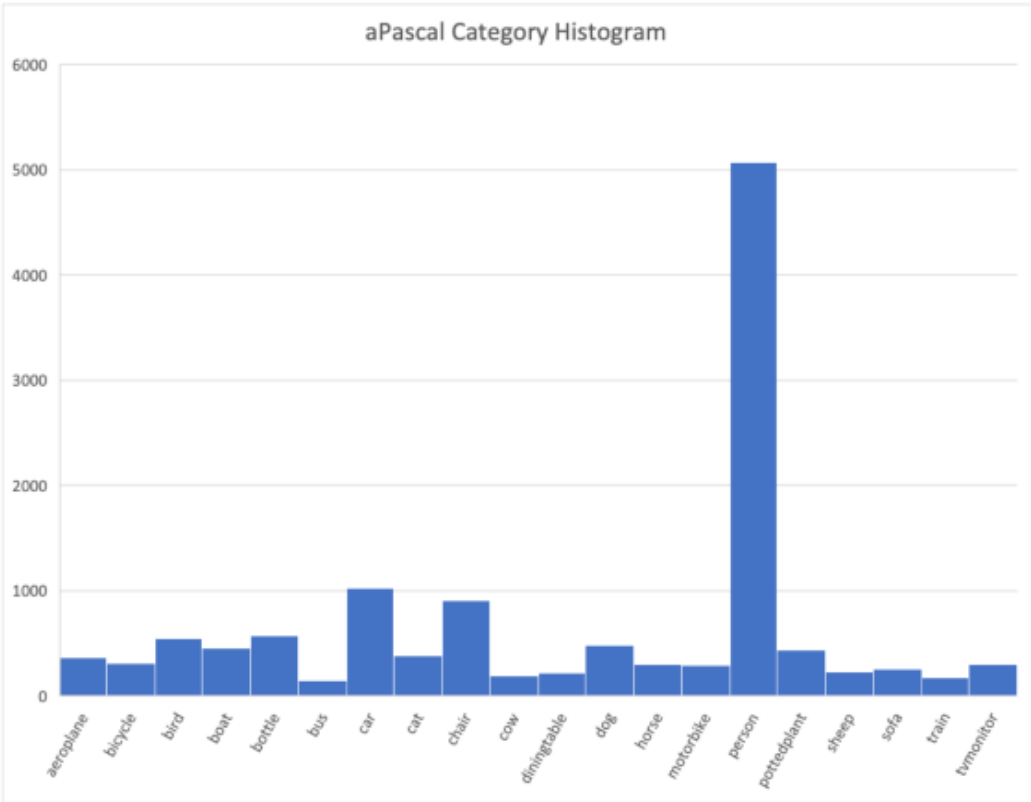


Figure 4.1: Histogram of aPascal class categories. There are 20 classes on Pascal VOC Dataset [4]. Person class has the maximum number of annotations.

A total of 12695 data were collected, 6340 of which were train set and 6355 were test set on Pascal VOC dataset [4]. The histogram of aPascal classes is shown in Figure 4.1.

We calculate the average of the 12695 point features, including training and test sets of the Pascal VOC dataset [4] based on class categories to get a $20x64$ feature matrix. This feature matrix is used by injecting into a model in our experiments.

The attribute names are as follows: *2D Boxy, 3D Boxy, Round, Vert Cyl, Horiz Cyl, Occluded, Tail, Beak, Head, Ear, Snout, Nose, Mouth, Hair, Face, Eye, Torso, Hand, Arm, Leg, Foot/Shoe, Wing, Propeller, Jet engine, Window, Row Wind, Wheel, Door, Headlight, Taillight, Side mirror, Exhaust, Pedal, Handlebars, Engine, Sail, Mast, Text, Label, Furn. Leg, Furn. Back, Furn. Seat, Furn. Arm, Horn, Rein, Saddle, Leaf, Flower, Stem/Trunk, Pot, Screen, Skin, Metal, Plastic, Wood, Cloth, Furry, Glass, Feather, Wool, Clear, Shiny, Vegetation, Leather*.

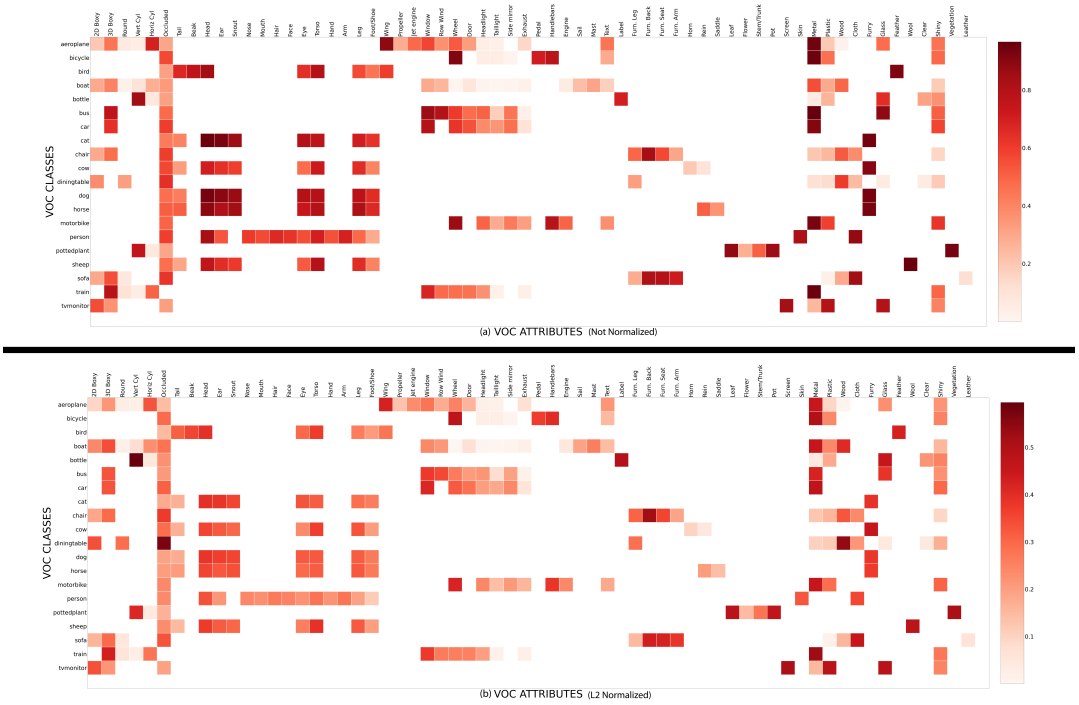


Figure 4.2: Attribute class heatmap based on Pascal VOC dataset [4] and aPascal attributes [16]. (a) Original attributes. Some attribute values are very high. (b) Attribute values are ℓ_2 normalized along with the feature matrix for all classes.

Attribute class heat map based on Pascal VOC dataset [4] class categories is shown in Figure 4.2. As shown in Figure 4.2 (a), some attribute values are very high. This is damaging to the network. Because of this reason, attribute values are ℓ_2 normalized along with the feature matrix for all classes for a better convergence of the network, shown in Figure 4.2 (b). Ablation study of the attributes normalization is examined in Section 7.1.2.

We examine aPascal attributes in detail by calculating the distance between categories. The most similar three categories are obtained with euclidean distance algorithm. Most similar 3 categories for each Pascal VOC dataset [4] category are shown in Table 4.1. As shown in the table, aeroplane category is *similar* to train, bus and car; cat category is similar to dog, cow and horse, as expected.

We use ℓ_2 normalized aPascal attributes for Pascal VOC dataset [4] experiments in this thesis.

4.3 GloVe

Pennington *et al.* [5] collect a bunch of vector representations for words, including Wikipedia, Gigaword, Common Crawl and Twitter datasets. The name of the vector representations is Global Vectors for Word Representation (GloVe). We get the values of word representations features for COCO dataset [3] categories to obtain 80×300 feature matrix.

Some classes from GloVe [5] does not exist in MSCOCO 2014 dataset [3]. We changed class names in GloVe dataset [5] compatible with MSCOCO 2014 dataset [3]. Class names mapping is given in Table 4.2.

We use ℓ_2 normalized GloVe embeddings for MSCOCO dataset [3] experiments in this thesis.

4.4 Google-News Word2vec

Word2vec can also be used as prior knowledge in weakly-supervised object detection. Mikolov *et al.* [6] has introduced google-news-300 word2vec. This work uses Google News corpus, containing 6B tokens for word2vec training. The higher the number of data, the higher the accuracy, but the data was limited to the 30K most frequently used words due to the high training time. An example word2vec of country-capital is shown in Figure 4.3.

Pascal VOC dataset [4] category word representations were selected from google-news-300 word2vec and created a 20×300 feature matrix. Some categories of the Pascal VOC dataset [4] is not present in google-news-300 word2vec, so we made a matching between dataset and word2vec categories. This matching is shown in Table 4.3. This feature matrix will be injected into models as prior knowledge. To obtain google-news-300 word2vec, gensim python package [137] was used, which is specialized in NLP (Natural Language Processing) and topic modeling.

We used google-news-300 word2vec matrix for Pascal VOC dataset [4] experiments in this thesis.

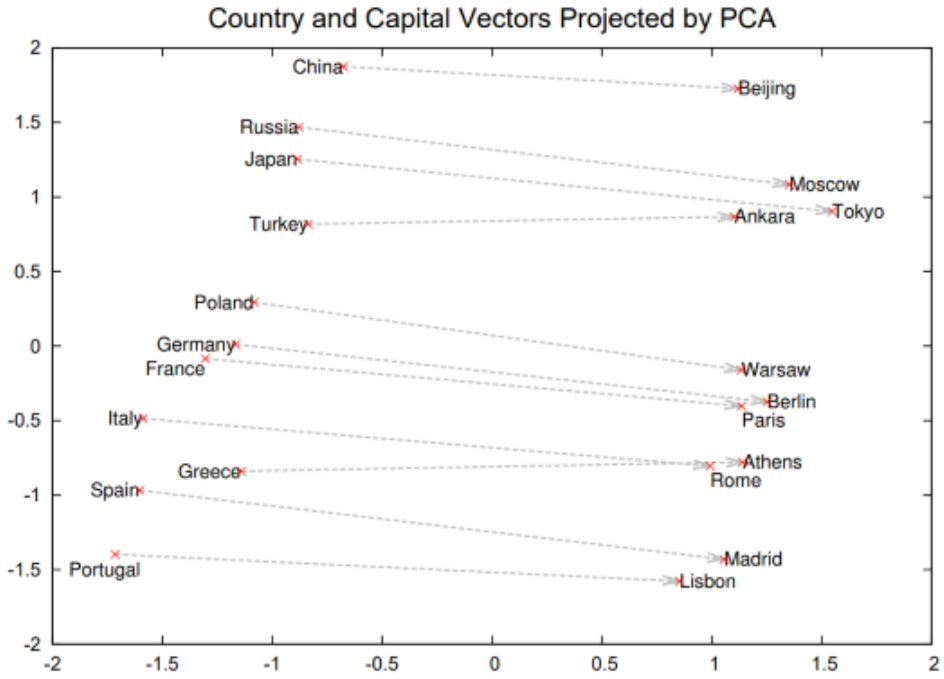


Figure 4.3: An example of word2vec from countries to capitals projected by PCA. For example $vector(France) - vector(Paris) + vector(Ankara)$ will be very close to $vector(Turkey)$ if the model is well trained. This figure is taken from Mikolov *et al.* [136].

Table 4.1: Most similar 3 categories for each category of Pascal VOC dataset [4]. Same type categories are similar to each other. For example *car* category is similar to *bus*, *train* and *aeroplane* categories.

Category	**Most Similar Categories**
aeroplane	train, bus, car
bicycle	motorbike, car, aeroplane
bird	dog, horse, cat
boat	train, aeroplane, car
bottle	tvmonitor, bus, pottedplant
bus	car, train, aeroplane
car	bus, train, motorbike
cat	dog, cow, horse
chair	sofa, diningtable, boat
cow	dog, cat, horse
diningtable	chair, boat, sofa
dog	cat, cow, horse
horse	dog, cat, cow
motorbike	bicycle, car, bus
person	dog, cat, sheep
pottedplant	bottle, diningtable, boat
sheep	dog, cat, horse
sofa	chair, diningtable, boat
train	bus, car, aeroplane
tvmonitor	boat, bus, bottle

Table 4.2: Class Names Mapping From MSCOCO [3] to GloVe [5] dataset. Some classes from GloVe [5] does not exist in MSCOCO 2014 dataset [3].

Class Name of COCO 2014 [3]	**Class Name of GloVe [5]**
sports ball	basketball
baseball glove	mitt
pottedplant	plant
tennis racket	racket

Table 4.3: Class Names Mapping From Pascal VOC [4] to google-news-300 [6]. Some classes from word representations does not exist in Pascal VOC dataset [4].

Class Name of Pascal VOC [4]	**Class Name of google-news-300 [6]**
aeroplane	plane
diningtable	table
pottedplant	plant
tvmonitor	tv

CHAPTER 5

WEAKLY-SUPERVISED OBJECT DETECTION FROM IMAGES USING PRIOR KNOWLEDGE

5.1 Overview

Weakly-supervised object detection models learn from binary image-level labels that represent whether an object instance of a specific class exists in an image or not. It reduces data preparation efforts for object detection problems by learning through image-level labels. Since collecting image-level labels without object locations (bounding boxes) is much easier than collecting bounding box annotations, weakly-supervised object detection methods are seen as more scalable than fully-supervised object detection methods, at least in principle.

The most apparent challenge in WSOD is the tendency to focus more on the discriminative areas than complete object instances. These challenges can be divided into three categories: missing instance; grouped instance; and part domination, shown in Figure 5.1. In this chapter, we explain our technical approaches towards improving WSOD through prior knowledge.

5.2 Approach

This thesis aims to address the WSOD weaknesses using prior knowledge of object classes based upon their textual descriptions, attributes and taxonomies. We take the widely-used WSDDN [1] approach as our baseline network and study prior knowledge guidance for WSOD on top of the WSDDN model.

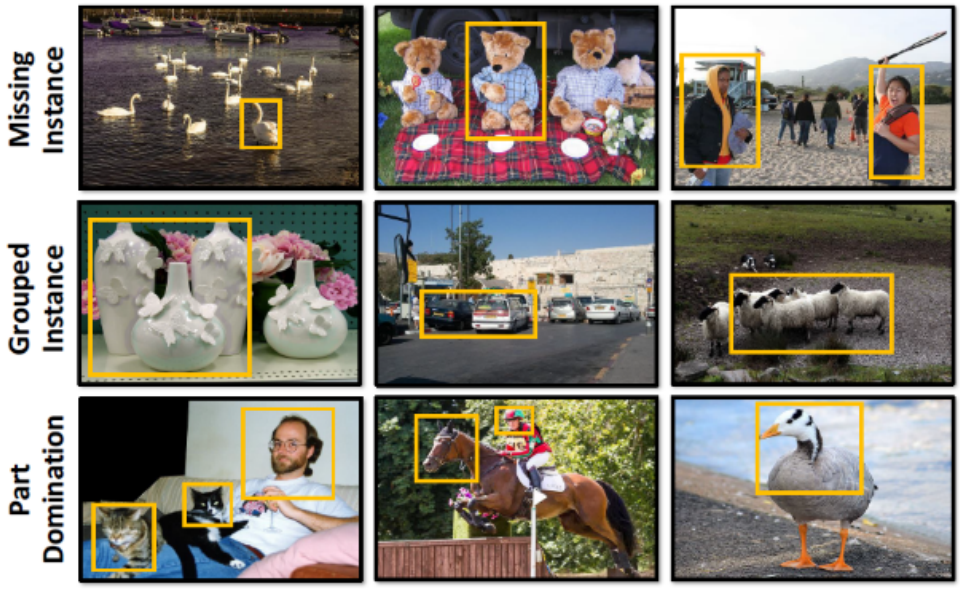


Figure 5.1: Main challenges in WSOD. **Missing instance:** Network does not consider distant or less salient objects. **Grouped instance:** Network can group instances belonging to the same category. **Part domination:** Network is more concerned with the discriminative areas such as the cat's face than its whole body. The figure is taken from Ren *et al.* [70]

In the following sections, we first summarize the WSDDN approach and then explain our WSDDN extension for introducing prior knowledge.

5.2.1 WSDDN (Weakly-Supervised Deep Detection Networks) [1]

In WSDDN, the weakly supervised detection task is divided into two sub-parts. The first one is the classification branch that assigns each region to a class. As a result, class scores ($\phi^c(x; R)$) are computed for each region, and these scores are normalized with softmax. Normalized scores for classification stream are denoted by $\sigma_{class}(x^c)$. The second part is the detection branch that picks the most promising windows in an image for each class. The probability distribution $\phi^d(x; R)$ is computed to compare regions over classes and normalized with softmax scores. Normalized scores for detection stream are denoted as $\sigma_{det}(x^d)$. Detection and recognition scores are

combined with Hadamard (element-wise) product:

$$x^{\mathcal{R}} = \sigma_{class}(x^c) \odot \sigma_{det}(x^d) \tag{5.1}$$

After region scores $x^{\mathcal{R}}$ are computed, the sum over regions is used to calculate image-level scores:

$$y_c = \sum_{r=1}^{|\mathcal{R}|} x_{cr}^{\mathcal{R}} \tag{5.2}$$

which results in a value within the range $[0, 1]$. By treating y_c as the image-level class probability and maximizing the log probability $\log y_c$, the WSOD model is then trained in an end-to-end manner.

5.2.2 Proposed Method

With region proposal method SSW [51], bounding boxes $b_i \in \mathbb{N}^4$ and the labels y_i are generated. The representation of each box b_i is denoted by $\phi(b_i) \in \mathbb{R}^{D_1}$. The corresponding semantic embeddings are denoted by $w_j \in \mathbb{R}^{D_2}$. The goal is to embed class attributes and image features in the same space.

For the region proposal method, we use Selective Search Windows (SSW) [51] and Multiscale Combinatorial Grouping (MCG) [138] algorithms. aPascal [16] class attributes are used for weakly-supervised object detection as prior knowledge. The training dataset has only one-hot image-level annotation as ground truth information in our setting.

We prepare class embeddings matrix ($W_j \in \mathbb{R}^{CxD_2}$) where C is class (category) count, D_2 is feature vector size for class embeddings. We multiply class embeddings matrix (w_j) with $\sigma_{class}(x^c) \in \mathbb{R}^{RxD_2}$ for classification stream and multiply class embeddings matrix (W_j) with $\sigma_{det}(x^d) \in \mathbb{R}^{RxD_2}$ for detection stream. New dimensions of $\sigma_{class}(x^c)$ and $\sigma_{det}(x^d)$ are $R \times C$.

Final scores of each region are obtained by taking element-wise product of $\sigma_{class}(x^c)$ and $\sigma_{det}(x^d)$. To get image-level scores, we sum the region scores x^R with Equation 5.3.

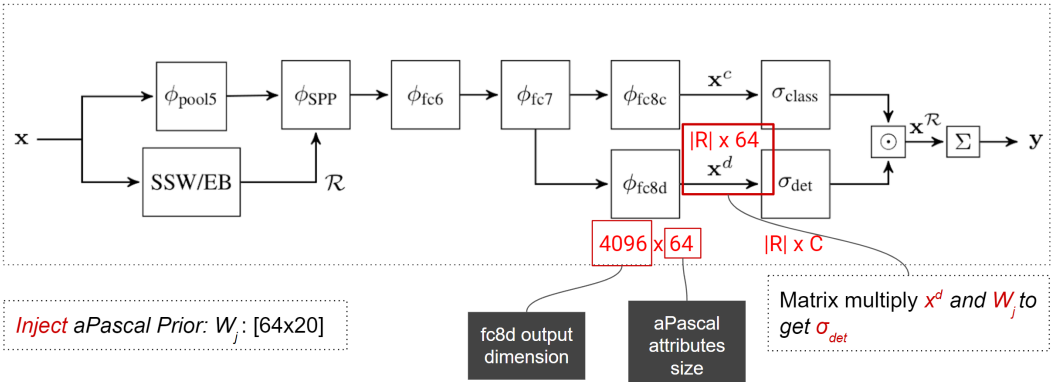


Figure 5.2: Our proposed method architecture for WSDDN [1] method. ϕ_{fc8c} and ϕ_{fc8d} denotes the last fully-connected layer for classification and detection branches respectively. We change the ϕ_{fc8c} and ϕ_{fc8d} shape from 4096×20 to 4096×64, where 64 is the aPascal [16] attributes prior feature size. We apply a matrix multiplication with x^d and W, where W is the aPascal [16] attributes prior to obtain σ_{det} and σ_{class} input.

$$y_c = \sum_{r=1}^{|R|} x_{cr}^R \tag{5.3}$$

Our proposed method architecture is shown in Figure 5.2. ϕ_{fc8c} and ϕ_{fc8d} denotes the last fully-connected layer for classification and detection branches respectively. We change the ϕ_{fc8c} and ϕ_{fc8d} shape from 4096×20 to 4096×64, where 64 is the aPascal [16] attributes prior feature size. We apply a matrix multiplication with x^d and W_j, where W_j is the aPascal [16] attributes prior to obtain σ_{det} and σ_{class} input. σ_{det} is the softmax of output of fully connected layer x^d. The shape of σ_{det} is $R \times C$, where R is the region proposals size, and C is the class size.

5.3 Summary

This chapter presents a modified version of Weakly-Supervised Deep Detection Network [1] architecture that incorporates prior knowledge in the form of class embeddings. The experiments are presented in Section 7.1.

CHAPTER 6

WEAKLY-SUPERVISED SEMANTIC SEGMENTATION FROM IMAGES USING PRIOR KNOWLEDGE

6.1 Overview

In various problems, bounding box predictions can be too coarse and producing pixel-level predictions can be desired. Recent fully-semantic segmentation methods achieve remarkable progress in computer vision. However, the state-of-the-art approaches rely on pixel-level annotations for training. In this chapter, we present our efforts towards leveraging existing prior knowledge to improve weakly supervised semantic segmentation models.

6.2 Baseline Method

Weakly-supervised semantic segmentation methods have emerged to reduce the annotation costs by enabling training with image-level labels. Most of the state-of-the-art methods on weakly-supervised semantic segmentation use CAM (Class Activation Maps) [42] based formulations over trained classification models to generate pseudo-ground truth pixel proposals to train fully-supervised semantic segmentation models such as DeepLab [14].

We use Puzzle-CAM [9] as the baseline method in our weakly-supervised semantic segmentation experiments. Puzzle-CAM [9] method, similar to many other methods, uses CAM (Class Activation Maps) [42] to generate ground truth pixel maps. These pixel maps are then used as ground truth on fully-supervised semantic segmentation models.

Figure 6.1: Weakly-supervised semantic segmentation models tend to focus on discriminative regions such as head of a cat instead of the whole body. Objects with a head such as person, cat, dog have this problem more because head is the most discriminative region to classify from other categories. The figure is taken from Kim *et al.* [139]

A main challenge in weakly-supervised segmentation is handling the tendency to focus only on discriminative regions in images, such as the head of a bird instead of the whole body. A typical problem is shown in Figure 6.1. To address this problem, Puzzle-CAM splits the image into multiple tiles and computes CAMs for each tile. The tile CAMs are then merged into a single CAM map. A second global CAM map is also generated from the original input image. The network of Puzzle-CAM [9] uses reconstruction loss ($\mathcal{L}_{re}$) between merged CAM and original CAM to reduce the dissimilarity using self-supervision. L1 loss is used for reconstruction loss. An example of CAM generation from the original image and tiled images are shown in Figure 6.2.

Notations of the Puzzle-CAM [9] method is as follows: In our summary of Puzzle-CAM, we use the following notation: I denotes the input image. F denotes the feature representation given by a ConvNet, e.g., ResNet-50 [53]. $G()$ is the Global

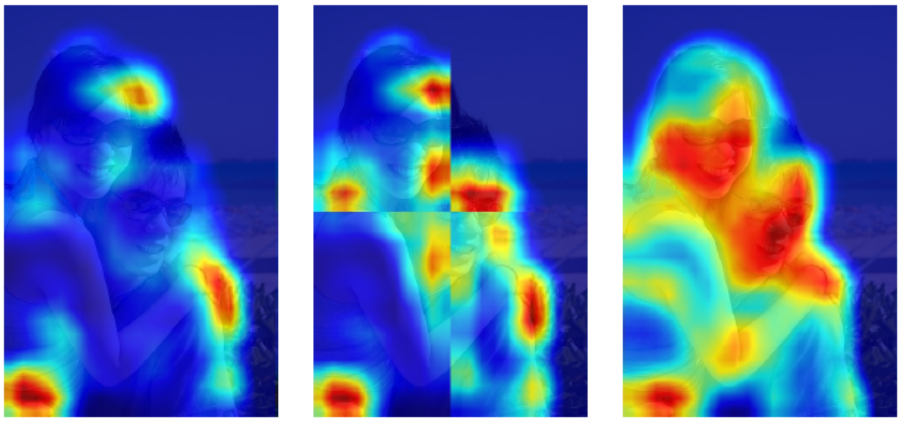


Figure 6.2: An example of Puzzle-CAM method CAM generation. (a) CAM is generated from the original image. CAM outputs do not cover the objects in the image. (b) CAMs are generated from tiled images and then merged into a single CAM. (c) Puzzle-CAM model CAM prediction. The figure is taken from Jo *et al.* [9].

Average Pooling (GAP) layer. θ is the classifier to be trained. A_c is CAM for class c.

$$A_c = \theta_c^T f. \tag{6.1}$$

A is the concatenated CAM of class CAMs A_c. A^s is the CAM of the original image. A^{re} is the CAM of the tiled images. Y denotes one-hot encoded ground truth vector from image-level annotations. $\hat{Y}$ is the class prediction vector, given by sigmoid of CAM for all classes. $\hat{Y}_t$ is $\hat{Y}$ for positive classes, $1 - \hat{Y}$ for negative classes, as also shown in Equation 6.5. $\hat{Y}^s$ is the prediction vector for original (A^s) CAM using GAP layer, as shown in Equation 6.2.

$$\hat{Y}^s = G(A^s) \tag{6.2}$$

$\hat{Y}^{re}$ is the prediction vector for tiled (A^{re}) CAM using GAP layer, as shown in Equation 6.3.

$$\hat{Y}^{re} = G(A^{re}) \tag{6.3}$$

$\mathcal{L}_{re}$ is the reconstruction loss between merged CAM from tiled images and CAM from the original image, where ℓ_1 loss is used. $\mathcal{L}_{cls}$ is the classification loss of the

original image, where multilabel soft margin loss is used. $\mathcal{L}_{p-cls}$ is the classification loss of the merged images, where again multilabel soft margin loss is used.

Input image (I) is fed into feature extractor $F()$ to generate feature map using equation 6.4.

$$f = F(I). \tag{6.4}$$

The CAM of a category c is generated by applying the weights of classifier θ to feature map of a class. Concatenation of class CAMs (A_c) is final CAM (A).

$\hat{Y}_t$ is used on loss calculations, which is a manipulation of $\hat{Y}$ such that returns $\hat{Y}$ for positive classes and $1 - \hat{Y}$ for negative classes, shown in Equation 6.5. Multi-label soft margin loss is used as classification loss shown in Equation 6.6.

$$\hat{Y_t} = \begin{cases} \hat{Y}, & \text{if } Y = 1 \\ 1 - \hat{Y}, & otherwise \end{cases} \tag{6.5}$$

$$\ell_{cls}(\hat{Y}, Y) = -\log(Y_t). \tag{6.6}$$

The classification loss of the original image and merged from tiled images are calculated as Equation 6.7 and Equation 6.8 respectively.

$$\mathcal{L}_{cls} = \ell_{cls}(\hat{Y}^s, Y) \tag{6.7}$$

$$\mathcal{L}_{p-cls} = \ell_{cls}(\hat{Y}^{re}, Y) \tag{6.8}$$

Reconstruction loss (L1) between merged CAM from tiled images and CAM from the original image is shown in Equation 6.9. This loss aims to reduce the dissimilarity between reconstruction CAM and original CAM.

$$\mathcal{L}_{re} = \|A^s - A^{re}\|_1 \tag{6.9}$$

The final loss of Puzzle-CAM [9] is shown in Equation 6.10. To change the weight of reconstruction loss ($\mathcal{L}_{re}$), α is used.

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{p-cls} + \alpha\mathcal{L}_{re} \tag{6.10}$$

Puzzle-CAM [9] architecture is shown in 2.16.

6.3 Proposed Method

We propose adding prior knowledge as class attributes (aPascal [16]) into the classification network (θ) to increase the quality of the pixel maps.

The baseline classifier uses ResNest [140] architecture as backbone for feature extraction from images. The features are then feed into 1×1 convolutional layer with 2048 input channels and 20 output channels (category size). The convolution layer output is fed into global average pooling (GAP) to get 1×20 logits. The output logits are used for classification loss with ground truth image labels.

We change the 1×1 convolutional layer with our proposed EmbedClassifier module which contains a 1×1 convolutional adaptation layer with 20 input channels (category size) and 64 output channels (attributes size) for aPascal [16] attributes as prior knowledge. The adaptation layer values are filled with kaiming normal initialization [141]. aPascal [16] attributes matrix is used as the weight of the 1×1 convolutional adaptation layer. aPascal [16] attribute gradients are fixed which is not changed during training the model.

6.4 Summary

Fully-supervised semantic segmentation models, such as [12, 30, 11], have gained important success in computer vision. However, to train a fully-supervised semantic segmentation model, a carefully annotated training dataset is needed. Annotating every pixel in an image is a long and tedious operation. Weakly-supervised segmentation models are trained with only image-level labels, which are easy to collect and

annotate compared to pixel labels. However, weakly-supervised semantic segmentation models tend to focus on discriminative regions of the objects, such as a dog's head, because of the lack of information about categories. To solve this problem, we propose to inject class attributes as prior knowledge into the classification network. The experimental results are presented and discussed in Section 7.2.

CHAPTER 7

EXPERIMENTS

This chapter includes detailed experiments and ablation studies for weakly-supervised object detection in Section 7.1 and weakly-supervised semantic segmentation in Section 7.2 for Pascal VOC dataset [4] and COCO 2014 dataset [3].

7.1 Weakly-supervised Object Detection From Images Using Prior Knowledge

Our approach on weakly-supervised object detection using prior knowledge is explained in Section 5.2.2. We make several experiments and ablation studies using the proposed approach on Pascal VOC dataset [4] and COCO 2014 dataset [3] in section 7.1.1 and 7.1.2, respectively.

7.1.1 Experiments

There is still a considerable gap between fully-supervised and weakly-supervised object detection results. This can be observed from the state-of-the-art fully supervised and weakly supervised object detection method results for the Pascal VOC 2007 [4] dataset shown in Table 7.1.

We start evaluation with PASCAL VOC 2007 [4] dataset and continue with MSCOCO 2014 [3] dataset during experiments. Mean Average Precision (mAP) metric is used as the performance measure. PyTorch [148] framework is used for development. An NVidia RTX2060 GPU is used for all the experiments.

We evaluate our model using the PASCAL VOC 2007 [4] dataset. PASCAL VOC

Table 7.1: State-of-the-art object detection method AP50 results on Pascal VOC 2007 [4] validation dataset. The first two rows show the fully-supervised object detection results, and the rest rows show the weakly-supervised object detection results. SSW: Selective Search Windows. EB: Edge Boxes. RPN: Region Proposal Network.

Method	**Proposal**	**mAP (0.50)**
Fast R-CNN [23]	SSW	0.669
Faster R-CNN [24]	RPN	**0.699**
WSDDN [1] (Baseline)	EB	0.348
OICR [10]	SSW	0.412
PCL [7] (Baseline)	SSW	0.435
SDCN [142]	SSW	0.502
C-MIL [143]	SSW	0.505
Yang *et al.* [144]	SSW	0.515
C-MIDN [145]	SSW	0.526
Pred Net [146]	SSW	0.529
WSOD2 [147]	SSW	0.536
Ren *et al.* [70]	SSW	**0.549**

2007 [4] dataset has 20 classes. Most of the Weakly-Supervised Object Detection (WSOD) methods [142, 90, 1] evaluate their results with the PASCAL VOC dataset [4]. We compare our result with WSDDN [1] model, which is our baseline.

For the region proposal method, Selective Search Windows (SSW) [51] is used. Region proposals are not computed during training and test instead, pre-computed Selective Search Windows (SSW) [51] region proposals are used. For the embedding model, aPascal [16] class attributes are used.

There are 20 classes in PASCAL VOC 2007 [4] dataset. We prepare the class embeddings matrix ($w_j \in \mathbb{R}^{CxD_2}$) where C is 20, D_2 is 64 for PASCAL VOC 2007 [4] dataset with aPascal [16] attributes. aPascal [16] attributes' feature dimension is 64 for all classes. We multiply class embeddings matrix (w_j) with $\sigma_{class}(x^c) \in \mathbb{R}^{RxD_2}$ for classification stream and multiply class embeddings matrix (w_j) with $\sigma_{det}(x^d) \in \mathbb{R}^{RxD_2}$. New dimensions of $\sigma_{class}(x^c)$ and $\sigma_{det}(x^d)$ is RxC. We use Adam [149] op-

Table 7.2: PASCAL VOC 2007 [4] evaluation results for WSDDN [1] and our proposed method. Mean Average Precision (mAP) metric with 0.5 IoU is used.

Method	**mAP (0.50)**
Base Network - w/o embedding [1]	0.3357
Ours - with embedding	**0.3868**

timizer with learning rate as $1e^{-5}$ and momentum as 0.9 and weight decay as $5e^{-4}$. We use AlexNet [99] as backbone and initialize the model with pre-trained weights on ImageNet [99]. Firstly, we evaluate the result with WSDDN [1] model without embeddings. Secondly, we add embeddings to the WSDDN model. The results show that, without embedding, model can converge up to 33%, but when we add embedding to the model, model can converge up to the 38% mAP with 140000 iterations. The results are given in Table 7.2.

We also try an experiment for zero-shot object detection with the aPascal [16] embedding model. PASCAL VOC 2007 [4] dataset is used for this experiment. We select the **aeroplane** class as a target category. The rest of the PASCAL VOC 2007 dataset classes are selected as source categories. There is no supervision for aeroplane class, including image-level labels. For the 19 source classes, image-level annotations are used as ground truth. aPascal [16] attributes are used for source and target classes in this experiment. The results show that with only class embeddings for target classes and with only class-level labels for source classes, average precision (AP) of the target category (aeroplane) increases up to 8% with 100000 iterations.

We evaluate our model with challenging MSCOCO 2014 [3] dataset. MSCOCO 2014 [3] dataset has 80 classes. MSCOCO 2014 [3] dataset is challenging because 41% of the dataset contains small (area $<32^2$) objects. As the region proposal method, the Multiscale Combinatorial Grouping (MCG) [138] method is selected because MCG tends to yield better results than Selective Search Windows (SSW) [51] on the MSCOCO [3] dataset. Pre-computed MCG proposals are used. As the embedding model, Global Vectors for Word Representation (GloVe) [5] is used. Some classes from GloVe do not exist in MSCOCO 2014 dataset. We change class names in GloVe [5] dataset compatible with MSCOCO 2014 [3] dataset. Class names mapping is

Table 7.3: Class Names Mapping From MSCOCO [3] to GloVe [5] dataset. Some classes from GloVe does not exist in MSCOCO 2014 dataset.

Class Names of COCO 2014 [3]	**Class Names of GloVe [5]**
sports ball	basketball
baseball glove	mitt
potted plant	plant
tennis racket	racket

Table 7.4: COCO 2014 [3] evaluation results for WSDDN [1] and our proposed method. Mean Average Precision (mAP) metric with 0.5 IoU is used.

Method	**mAP (0.50)**
Base Network - w/o embedding [1]	0.011
Ours - with embedding	**0.12**

given in Table 7.3.

We first evaluate the baseline WSDDN [1] model without embedding. Second, we add our embeddings into the WSDDN model. The results are given in Table 7.4. These results suggest that, without embedding, model can not converge, but when we add embedding to the model, model can converge up to the 12% mAP with 260000 iterations.

Correct detection results are shown in Figure 7.1. Results are computed with our proposed model on PASCAL VOC 2007 [4] test set. aPascal [16] class attributes are normalized. Best mean average precision (mAP) is 38%. False detection results are shown in Figure 7.2.

We now change our baseline model with PCL (Proposal Cluster Learning) [7] to inject prior knowledge. PCL [7] model is also using WSDDN [1] architecture as the baseline, explained in detail in section 2.1.2. Pascal VOC 2007 dataset [4] and SSW (Selective search windows) [51] are used for training. We train the baseline network (PCL) with the Pascal VOC 2007 dataset [4] and get 0.480 mAP score, shown in

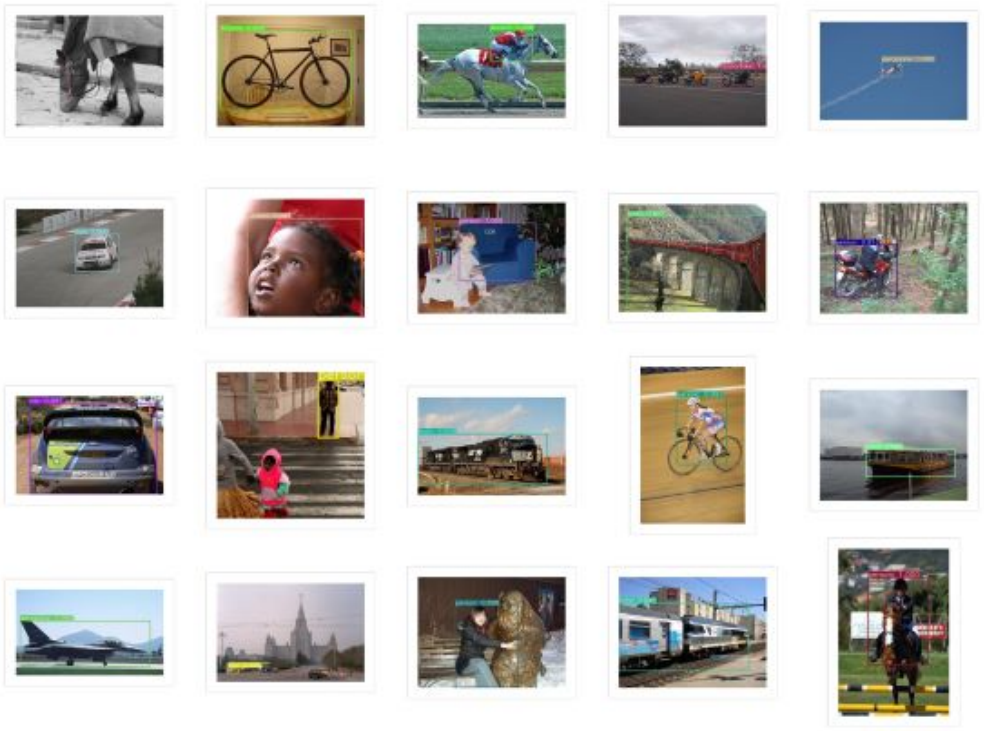


Figure 7.1: Examples of correct detection results with our model. Detection results are based on PASCAL VOC 2007 [4] dataset. aPascal [16] attributes are used and L2 normalized.

Table 7.5.

We inject the aPascal [16] attributes as prior into every refinement layer for classification and detection branches. We first train the PCL [7] model with only aPascal [16] attributes as prior but does not pass the baseline score. Then, we concatenate aPascal [16] attributes with Google-News word2vec and train the model for a better evaluation score than the baseline. The impact of the aPascal [16] attributes and word2vec is shown in Table 7.5.

We train the baseline network (PCL [7]) with COCO 2014 dataset [3] and get 0.2107 AP50 score. We inject GloVe [5] prior into every refinement layer for classification and detection branches for the PCL [7] model. The comparison of the AP50 scores of baseline and our proposed method is shown in Table 7.6.

We compare the number of parameters for the baseline method and the proposed

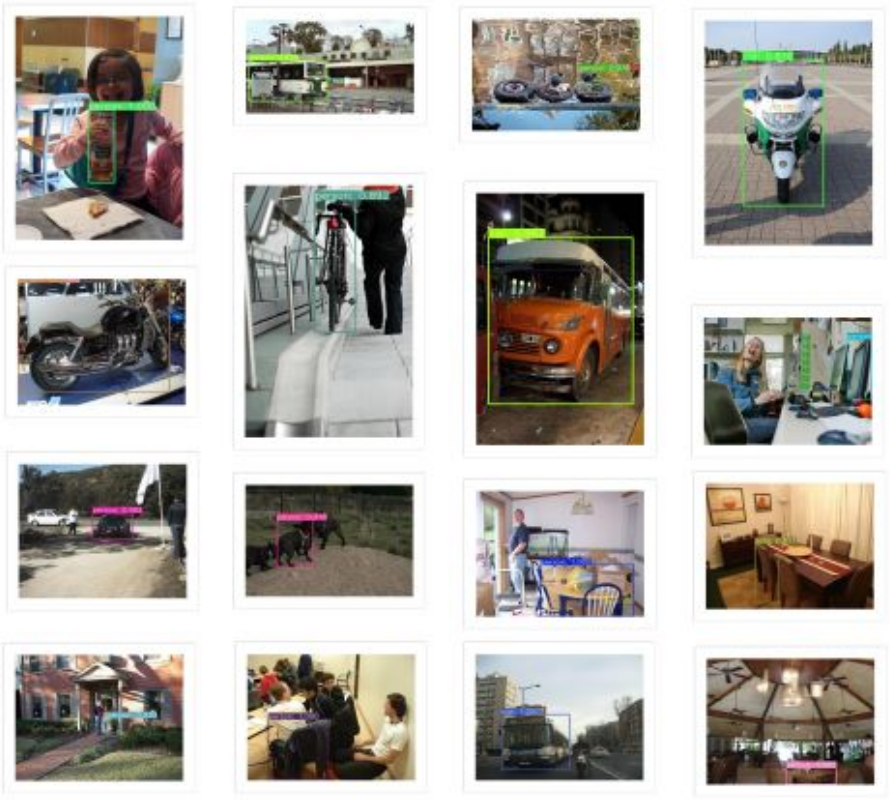


Figure 7.2: Examples of false detection results with our model. Detection results are based on PASCAL VOC 2007 [4] dataset. aPascal [16] attributes are used and L2 normalized.

method in Table 7.7. The results show that the difference in the number of parameters between the proposed method and the baseline method is negligible.

7.1.2 Ablation Study

aPascal [16] class attributes are L2 normalized to increase model performance and stability. aPascal [16] class attributes matrix is denoted as $w_j \in \mathbb{R}^{CxD_2}$ where C is 20, D_2 is 64 for PASCAL VOC 2007 [4] dataset.

Class embeddings are L2 normalized along the feature dimensions for all classes. Norm vector ($|w_j| \in \mathbb{R}^{CxD_2}$) of the feature vector is calculated via Equation 7.1.

Table 7.5: Pascal VOC 2007 [4] evaluation results with PCL [7] baseline method. We inject only aPascal [16] attributes and only Google-news word2vec as prior into the model and does not pass the baseline score. We concatenate aPascal [16] and Google-news word2vec and inject into the model and pass the baseline score. Mean Average Precision (mAP) metric with 0.5 IoU is used.

Method	**aPascal [16]**	**Google-news word2vec**	**mAP**
PCL [7] (Baseline)			0.480
Ours	✓		0.404
Ours		✓	0.471
Ours	✓	✓	0.481

Table 7.6: COCO 2014 dataset [3] evaluation results with PCL [7] baseline method. We inject GloVe [5] as prior into the model. Mean Average Precision (mAP) metric with 0.5 IoU is used.

Method	**AP50**
PCL [7] (Baseline)	0.210
Ours	0.218

Normalized class embeddings are obtained by dividing the feature vector by the norm vector in Equation 7.2.

$$|w_j| = \sum_{i}^{D_2} \sqrt{(w_{ji})^2} \tag{7.1}$$

$$w_j = \frac{w_j}{|w_j|} \tag{7.2}$$

First of all, we train our model with unnormalized class embeddings. Then, we L2 normalize class embeddings to compare the results. Results are shown in Table 7.8. Normalizing aPascal [16] class attributes improves the model performance.

Table 7.7: Comparison of the number of parameters between baseline and our proposed method. We inject aPascal [16] attributes to WSDDN [1] and PCL [7] models and show the number of parameters.

Method	Backbone	Baseline	Inject Prior (Proposed Method)
WSDDN [1]	VGG16	138,521,424	138,881,960
PCL [7]	VGG16	135,112,720	136,177,940

Table 7.8: PASCAL VOC 2007 [4] evaluation results with normalization. Mean Average Precision (mAP) metric with 0.5 IoU is used. The results are compared with normalized embeddings factor. Normalizing aPascal [16] class attributes improves the model performance.

Method	mAP (0.50)
Embeddings Not Normalized	0.375
Embeddings Normalized	0.386

7.2 Weakly-supervised Semantic Segmentation From Images Using Prior Knowledge

Our approach on weakly-supervised semantic segmentation using prior knowledge is explained in Section 6.3. We make several experiments and ablation studies on Pascal VOC dataset [4] in section 7.2.1 and 7.2.2, respectively, to investigate the proposed approach.

7.2.1 Experiments

There is still a considerable gap between fully-supervised and weakly-supervised semantic segmentation results. State-of-the-art semantic segmentation method results, with fully or weakly supervised training, on Pascal VOC 2012 [8] dataset are shown in Table 7.9. The differences highlight the performance gap.

We evaluate our proposed method with Pascal VOC 2012 [8] dataset. We follow the

Table 7.9: State-of-the-art semantic segmentation mIoU results on Pascal VOC 2012 [8] validation dataset. The first two rows show the fully-supervised semantic segmentation results, and the following rows show the weakly-supervised semantic segmentation results. P: Pixel-level labels. I: Image-level labels. S: External saliency methods.

Method	**Backbone**	**Supervision**	**mIoU**
DFN [150]	ResNet-101	P	80.60
ExFuse [151]	ResNeXt-131	P	**85.8**
DSRG [46]	ResNet-101	$I + S$	61.4
AffinityNet [43]	Wide-ResNet-38	I	61.7
SeeNet [45]	ResNet-101	$I + S$	63.1
IRNet [43]	ResNet-50	I	63.5
ICD [152]	ResNet-101	I	64.1
SEAM [15]	Wide-ResNet-38	I	64.5
FickleNet [44]	ResNet-101	$I + S$	64.9
Puzzle-CAM [9] (Baseline)	ResNeSt-101	I	**66.9**

same dataset images and annotations with Puzzle-CAM [9] method. Mean Intersection over Union (mIoU) is used as a performance metric. Details of the datasets and metrics are examined in Chapter 3. PyTorch [148] deep learning framework is used for development. We use an NVidia Tesla A100 GPU in all experiments.

We use aPascal [16] class attributes as prior for the classification network. We apply L2 normalization to aPascal [16] class attributes before training. Details of aPascal [16] attributes are examined in section 4.2.

We first reproduce Puzzle-CAM [9] results by training ResNeSt-101 [140] backbone. Jo *et al.* [9] achieve 66.9 mIoU on Pascal VOC 2012 [8] validation dataset using ResNeSt-101 [140] backbone. We train with the same settings and achieve 65.4 mIoU on Pascal VOC 2012 [8] validation dataset using ResNeSt-101 [140] backbone. As a baseline result, 65.4 mIoU is used on the experiments.

We train the classification network with L2 normalized aPascal [16] class attributes.

Table 7.10: PASCAL VOC 2012 [8] evaluation results with Puzzle-CAM [9] baseline method. Mean Intersection over Union (mIoU) metric is used. Our proposed method increases the mIoU score compared to the baseline method. We concatenate the identity matrix of categories to aPascal [16] class attributes (concat identity), which increases the feature size from 64 to 84. Zero values are replaced with -0.1 on aPascal [16] class attributes (zeros replaced with -0.1).

Method	**Backbone**	**mIoU**
Baseline (Puzzle-CAM [9])	ResNeSt-101	65.478
Ours (concat identity)	ResNeSt-101	65.880
Ours (zeros replaced with -0.1)	ResNeSt-101	66.887

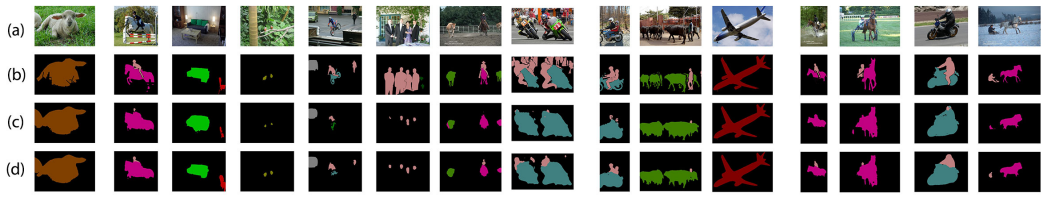


Figure 7.3: Semantic segmentation results from Pascal VOC dataset [8] example images. (a) Original image. (b) Ground truth segmentation maps. (c) Baseline (Puzzle-CAM [9]) model prediction. (d) Our proposed method model prediction. The original images (a) are taken from Pascal VOC 2012 Dataset [8].

This training does not improve the mIoU score any further. Then, we replace zeros with -0.1 on aPascal [16] class attributes and train the classification network. We also concatenate category identity matrix (20×20) with aPascal [16] class attributes (20×64) to create final (20×84) attributes and train the classification network. Our proposed method increases the mIoU score after both the zeros replacement and concatenated identity matrix, shown in Table 7.10.

Semantic segmentation results from example images are shown in Figure 7.3.

We compare our method with the baseline method on category scores. Category based IoUs are compared in Table 7.11. Our proposed method increases the IoU score for some categories such as aeroplane, bird, boat, bus, sheep and mIoU score. However,

Table 7.11: Class-based comparison of PASCAL VOC 2012 [8] dataset. Class-based scores are compared with the baseline method (Puzzle-CAM [9]). Intersection over Union (IoU) metric is used for categories. Our proposed method increases the IoU score compared to the baseline method for some categories. For example, bird IoU is increased from 81.43 to 89.32 with our proposed method.

Method	**background**	**aeroplane**	**bicycle**	**bird**	**boat**	**bottle**	**bus**	**car**	**cat**	**chair**	**cow**	**diningtable**	**dog**	**horse**	**motorbike**	**person**	**pottedplant**	**sheep**	**sofa**	**train**	**tvmonitor**	**mIoU**
Baseline	87.20	79.53	33.58	81.43	44.40	73.46	84.77	80.05	90.93	24.77	87.94	44.40	88.20	82.42	75.17	37.74	57.35	83.59	41.56	51.32	45.12	65.47
Ours	87.64	81.76	32.48	89.31	64.76	70.86	86.44	75.80	87.37	31.04	85.55	48.57	81.75	80.53	73.33	22.51	54.80	86.75	38.05	61.81	63.42	66.88

Table 7.12: Comparison of the number of parameters between baseline and our proposed method. We inject aPascal [16] attributes to Puzzle-CAM [9] model and show the number of parameters.

Method	**Backbone**	**Baseline**	**Inject Prior (Proposed Method)**
Puzzle-CAM [9]	ResneSt101	25,475,200	25,565,376

some category IoU scores are decreased, such as dog, sofa, bottle, person. There is room for improvement in our proposed method.

We generate a confusion matrix for categories to understand which category is predicted as which category. We compare all the pixel categories with the ground truth and count the values. Person class is generally predicted as background in both the baseline and proposed methods. The confusion matrix for the Pascal VOC dataset [8] categories comparison is shown in Figure 7.4. Since all pixels categories for the images are almost background class, background-background confusion value is set to zero for a better plot.

We compare the number of parameters used in the baseline and the proposed method in Table 7.12. The results show that the difference in the number of parameters between the two methods is negligible.

Table 7.13: PASCAL VOC 2012 [8] evaluation results for zeros replacements. Mean Intersection over Union (mIoU) metric is used. Our proposed method increases the mIoU score compared to the baseline method when we replace zeros with -0.5 in aPascal [16] attributes. When we replace zeros with -0.5 in aPascal [16] attributes, the mIoU is decreased.

Method	**Backbone**	**mIoU**
Baseline (Puzzle-CAM [9])	ResNeSt-101	65.478
Ours (zeros replaced with -0.1)	ResNeSt-101	66.887
Ours (zeros replaced with -0.5)	ResNeSt-101	17.483

7.2.2 Ablation Study

We replace the zeros in aPascal [16] attributes with -0.1 for a better convergence of the model. The results are shown in Table 7.10. This method improves the mIoU result. We replace the zeros in aPascal [16] attributes with -0.5 and the result is not promising. The comparison of the zeros replacements is shown in Table 7.13. Replacement with -0.5 does not give a promising result because we normalize the aPascal [16] attributes with L2 and the network does not update the aPascal [16] gradients, so the network may not converge with high attribute values.

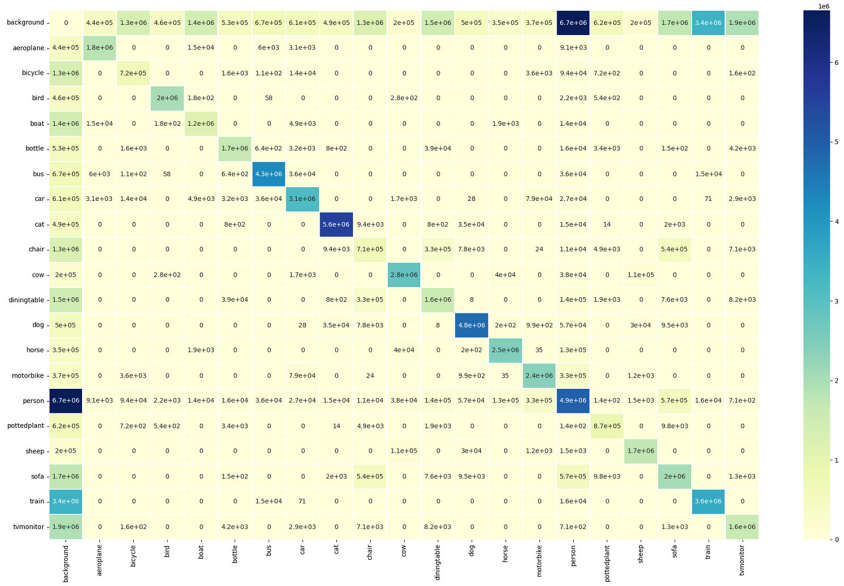
7.3 Summary

In this chapter, we make several experiments for weakly-supervised object detection and weakly-supervised semantic segmentation using Pascal VOC dataset [4] and COCO 2014 dataset [3]. We inject aPascal [16] attributes, Google-news word2vec and GloVe [5] as prior knowledge into network.

We have obtained clear evidence that our proposed method increases the mean average precision (mAP) metric for weakly-supervised object detection using prior knowledge. We also conduct experiments for zero-shot detection and get 8% mAP on PASCAL VOC 2007 dataset [4] for the aeroplane class. The results for weakly-supervised semantic segmentation show that our proposed method increases the mIoU

score compared to the baseline method, i.e., Puzzle-CAM [9].

(a)



(b)

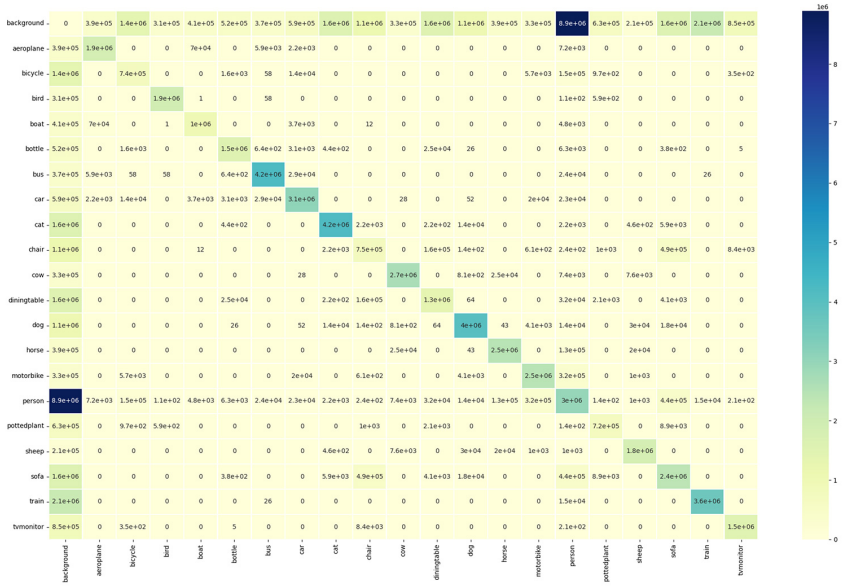


Figure 7.4: Confusion matrix for Pascal VOC dataset [8] categories. (a) Confusion matrix for baseline method (Puzzle-CAM [9]). (b) Confusion matrix for our proposed method. Background-background confusion value is set to zero. Person class is generally predicted as background in both the baseline and proposed methods.

CHAPTER 8

CONCLUSION

Fully-supervised object detection and semantic segmentation methods on computer vision gained tremendous success in recent years. Training fully-supervised models require fully annotated datasets. Annotating a dataset is a labor-intensive and tedious task that humans do. For one instance, annotating pixels takes 78 seconds per image [32]. Weakly-supervised methods solve this annotation problem by training models with only weak labels such as image-level, points and scribbles annotations. However, weakly-supervised model predictions tend to focus on discriminative regions of the objects, such as a head of a bird, which suggests that pure image-level label-based training can be insufficient for accurate training. We propose to add prior knowledge about object categories such as attributes (has hair, is plastic) into the model to reduce the shortcomings of weak annotations. We evaluate our proposed method on object detection models with Pascal VOC 2007 [4] and COCO 2014 [3] datasets and semantic segmentation models with Pascal VOC 2012 [8] dataset. Experiments show that our proposed method increases the baseline scores of weakly-supervised object detection and semantic segmentation models, with large improvements in various classes.

REFERENCES

[1] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2846–2854, 2016.

[2] 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*, vol. 88, no. 2, pp. 303–338, 2010.

[3] 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*, pp. 740–755, Springer, 2014.

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

[5] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[7] P. Tang, X. Wang, S. Bai, W. Shen, X. Bai, W. Liu, and A. Yuille, "Pcl: Proposal cluster learning for weakly supervised object detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 1, pp. 176–191, 2018.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes

Challenge 2012 (VOC2012) Results." http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[9] S. Jo and I.-J. Yu, "Puzzle-cam: Improved localization via matching partial and full features," in *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 639–643, IEEE, 2021.

[10] P. Tang, X. Wang, X. Bai, and W. Liu, "Multiple instance detection network with online instance classifier refinement," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2843–2851, 2017.

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

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[13] 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 conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[14] 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*, vol. 40, no. 4, pp. 834–848, 2017.

[15] 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*, pp. 12275–12284, 2020.

[16] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 1778–1785, IEEE, 2009.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[18] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.

[19] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[20] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[21] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9627–9636, 2019.

[22] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[23] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[25] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, pp. 379–387, 2016.

[26] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6154–6162, 2018.

[27] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[28] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra r-cnn: Towards balanced learning for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 821–830, 2019.

[29] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[30] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pp. 173–190, Springer, 2020.

[31] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*, pp. 424–432, Springer, 2016.

[32] 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*, pp. 549–565, Springer, 2016.

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[34] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," *Advances in neural information processing systems*, pp. 570–576, 1998.

[35] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997.

[36] H. Bilen, M. Pedersoli, and T. Tuytelaars, "Weakly supervised object detection with convex clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1081–1089, 2015.

[37] R. G. Cinbis, J. Verbeek, and C. Schmid, "Weakly supervised object localization with multi-fold multiple instance learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 1, pp. 189–203, 2016.

[38] V. Kantorov, M. Oquab, M. Cho, and I. Laptev, "Contextlocnet: Context-aware deep network models for weakly supervised localization," in *European Conference on Computer Vision*, pp. 350–365, Springer, 2016.

[39] H. O. Song, Y. J. Lee, S. Jegelka, and T. Darrell, "Weakly-supervised discovery of visual pattern configurations," in *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 1*, pp. 1637–1645, 2014.

[40] C. Wang, W. Ren, K. Huang, and T. Tan, "Weakly supervised object localization with latent category learning," in *European Conference on Computer Vision*, pp. 431–445, Springer, 2014.

[41] X. Wang, Z. Zhu, C. Yao, and X. Bai, "Relaxed multiple-instance svm with application to object discovery," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1224–1232, 2015.

[42] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

[43] J. Ahn and S. Kwak, "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4981–4990, 2018.

[44] 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*, pp. 5267–5276, 2019.

[45] Q. Hou, P.-T. Jiang, Y. Wei, and M.-M. Cheng, "Self-erasing network for integral object attention," *arXiv preprint arXiv:1810.09821*, 2018.

[46] 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 Conference on Computer Vision and Pattern Recognition*, pp. 7014–7023, 2018.

[47] 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*, pp. 2209–2218, 2019.

[48] Z. Ji and O. Veksler, "Weakly supervised semantic segmentation: From box to tag and back," 2021.

[49] 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 conference on computer vision and pattern recognition*, pp. 876–885, 2017.

[50] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3159–3167, 2016.

[51] 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.

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

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[54] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

[55] 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, pp. 580–587, 2014.

[56] 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*, pp. 21–37, Springer, 2016.

[57] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10781–10790, 2020.

[58] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "Reppoints: Point set representation for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9657–9666, 2019.

[59] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6569–6578, 2019.

[60] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 734–750, 2018.

[61] S. Lang, F. Ventola, and K. Kersting, "Dafne: A one-stage anchor-free deep model for oriented object detection," *arXiv preprint arXiv:2109.06148*, 2021.

[62] W. Ren, K. Huang, D. Tao, and T. Tan, "Weakly supervised large scale object localization with multiple instance learning and bag splitting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 405–416, 2015.

[63] O. Chum and A. Zisserman, "An exemplar model for learning object classes," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2007.

[64] T. Deselaers, B. Alexe, and V. Ferrari, "Weakly supervised localization and learning with generic knowledge," *International journal of computer vision*, vol. 100, no. 3, pp. 275–293, 2012.

[65] M. Pandey and S. Lazebnik, "Scene recognition and weakly supervised object localization with deformable part-based models," in *2011 international conference on computer vision*, pp. 1307–1314, IEEE, 2011.

[66] Z. Shi, T. M. Hospedales, and T. Xiang, "Bayesian joint modelling for object localisation in weakly labelled images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 10, pp. 1959–1972, 2015.

[67] H. O. Song, R. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, and T. Darrell, "On learning to localize objects with minimal supervision," in *International Conference on Machine Learning*, pp. 1611–1619, PMLR, 2014.

[68] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[69] P. Tang, X. Wang, Z. Huang, X. Bai, and W. Liu, "Deep patch learning for weakly supervised object classification and discovery," *Pattern Recognition*, vol. 71, pp. 446–459, 2017.

[70] Z. Ren, Z. Yu, X. Yang, M.-Y. Liu, Y. J. Lee, A. G. Schwing, and J. Kautz, "Instance-aware, context-focused, and memory-efficient weakly supervised object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10598–10607, 2020.

[71] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. Van Gool, "Weakly supervised cascaded convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 914–922, 2017.

[72] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European conference on computer vision*, pp. 391–405, Springer, 2014.

[73] R. Gokberk Cinbis, J. Verbeek, and C. Schmid, "Multi-fold mil training for weakly supervised object localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2409–2416, 2014.

[74] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International journal of computer vision*, vol. 105, no. 3, pp. 222–245, 2013.

[75] M. Blaschko, A. Vedaldi, and A. Zisserman, "Simultaneous object detection and ranking with weak supervision," in *Advances in neural information processing systems*, pp. 235–243, Citeseer, 2010.

[76] T. Deselaers, B. Alexe, and V. Ferrari, "Localizing objects while learning their appearance," in *European conference on computer vision*, pp. 452–466, Springer, 2010.

[77] M. H. Nguyen, L. Torresani, F. De La Torre, and C. Rother, "Weakly supervised discriminative localization and classification: a joint learning process," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 1925–1932, IEEE, 2009.

[78] O. Russakovsky, Y. Lin, K. Yu, and L. Fei-Fei, "Object-centric spatial pooling for image classification," in *European conference on computer vision*, pp. 1–15, Springer, 2012.

[79] K. K. Singh, S. Divvala, A. Farhadi, and Y. J. Lee, "Dock: Detecting objects by transferring common-sense knowledge," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 492–508, 2018.

[80] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, and L. Zhang, "Dynamic head: Unifying object detection heads with attentions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7373–7382, 2021.

[81] T. Liang, X. Chu, Y. Liu, Y. Wang, Z. Tang, W. Chu, J. Chen, and H. Ling, "Cbnetv2: A composite backbone network architecture for object detection," *arXiv preprint arXiv:2107.00420*, 2021.

[82] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao, "Focal self-attention for local-global interactions in vision transformers," *arXiv preprint arXiv:2107.00641*, 2021.

[83] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *arXiv preprint arXiv:2103.14030*, 2021.

[84] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-yolov4: Scaling cross stage partial network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13029–13038, 2021.

[85] J. Cao, Y. Pang, J. Han, and X. Li, "Hierarchical shot detector," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9705–9714, 2019.

[86] H. Li, B. Dai, S. Shi, W. Ouyang, and X. Wang, "Feature intertwiner for object detection," in *International Conference on Learning Representations*, 2018.

[87] B. Singh, M. Najibi, and L. S. Davis, "SNIPER: Efficient multi-scale training," *NeurIPS*, 2018.

[88] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2918–2928, 2021.

[89] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4203–4212, 2018.

[90] J. Hoffman, S. Guadarrama, E. S. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko, "Lsda: Large scale detection through adaptation," *Advances in neural information processing systems*, vol. 27, pp. 3536–3544, 2014.

[91] Y. Tang, J. Wang, B. Gao, E. Dellandréa, R. Gaizauskas, and L. Chen, "Large scale semi-supervised object detection using visual and semantic knowledge transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2119–2128, 2016.

[92] 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.

[93] N. Plath, M. Toussaint, and S. Nakajima, "Multi-class image segmentation using conditional random fields and global classification," in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 817–824, 2009.

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

[95] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[96] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. V. Le, "Rethinking pre-training and self-training," *arXiv preprint arXiv:2006.06882*, 2020.

[97] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

[98] 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*, pp. 3431–3440, 2015.

[99] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[100] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," *arXiv preprint arXiv:1506.04579*, 2015.

[101] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.

[102] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pp. 3–11, Springer, 2018.

[103] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 fourth international conference on 3D vision (3DV)*, pp. 565–571, IEEE, 2016.

[104] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao, "Adaptive pyramid context network for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7519–7528, 2019.

[105] D. Lin, Y. Ji, D. Lischinski, D. Cohen-Or, and H. Huang, "Multi-scale context intertwining for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 603–619, 2018.

[106] G. Li, Y. Xie, L. Lin, and Y. Yu, "Instance-level salient object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2386–2395, 2017.

[107] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[108] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

[109] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

[110] F. Visin, M. Ciccone, A. Romero, K. Kastner, K. Cho, Y. Bengio, M. Matteucci, and A. Courville, "Reseg: A recurrent neural network-based model for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 41–48, 2016.

[111] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville, and Y. Bengio, "Renet: A recurrent neural network based alternative to convolutional networks," *arXiv preprint arXiv:1505.00393*, 2015.

[112] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan, "Semantic object parsing with graph lstm," in *European Conference on Computer Vision*, pp. 125–143, Springer, 2016.

[113] Q. Huang, C. Xia, C. Wu, S. Li, Y. Wang, Y. Song, and C.-C. J. Kuo, "Semantic segmentation with reverse attention," *arXiv preprint arXiv:1707.06426*, 2017.

[114] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," *arXiv preprint arXiv:1805.10180*, 2018.

[115] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," *arXiv preprint arXiv:1611.08408*, 2016.

[116] N. Souly, C. Spampinato, and M. Shah, "Semi supervised semantic segmentation using generative adversarial network," in *Proceedings of the IEEE international conference on computer vision*, pp. 5688–5696, 2017.

[117] Y. Xue, T. Xu, H. Zhang, L. R. Long, and X. Huang, "Segan: Adversarial network with multi-scale l 1 loss for medical image segmentation," *Neuroinformatics*, vol. 16, no. 3, pp. 383–392, 2018.

[118] 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 conference on computer vision and pattern recognition*, pp. 1568–1576, 2017.

[119] 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 Conference on Computer Vision and Pattern Recognition*, pp. 7268–7277, 2018.

[120] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[121] 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*, pp. 4071–4080, 2021.

[122] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[123] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[124] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 524–531, IEEE, 2005.

[125] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 2169–2178, IEEE, 2006.

[126] A. Birhane and V. U. Prabhu, "Large image datasets: A pyrrhic win for computer vision?," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1536–1546, IEEE, 2021.

[127] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492, IEEE, 2010.

[128] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *2004 conference on computer vision and pattern recognition workshop*, pp. 178–178, IEEE, 2004.

[129] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories,"

IEEE transactions on pattern analysis and machine intelligence, vol. 28, no. 4, pp. 594–611, 2006.

[130] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2011.

[131] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 304–311, IEEE, 2009.

[132] 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*, vol. 128, no. 7, pp. 1956–1981, 2020.

[133] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, *et al.*, "Deep learning for generic object detection," *A Survey [J].*, 2018.

[134] P. J. Etude, "comparative de la distribution florale dans une portion des alpes et des jura," *Bull. Soc. Vaud. Sci. Nat*, vol. 37, p. 547, 1901.

[135] T. A. Sorensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons," *Biol. Skar.*, vol. 5, pp. 1–34, 1948.

[136] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[137] R. Rehurek and P. Sojka, "Gensim–python framework for vector space modelling," *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, vol. 3, no. 2, 2011.

[138] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Computer Vision and Pattern Recognition*, 2014.

[139] B. Kim, S. Han, and J. Kim, "Discriminative region suppression for weakly-supervised semantic segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 1754–1761, 2021.

[140] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, *et al.*, "Resnest: Split-attention networks," *arXiv preprint arXiv:2004.08955*, 2020.

[141] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

[142] X. Li, M. Kan, S. Shan, and X. Chen, "Weakly supervised object detection with segmentation collaboration," *arXiv preprint arXiv:1904.00551*, 2019.

[143] F. Wan, C. Liu, W. Ke, X. Ji, J. Jiao, and Q. Ye, "C-mil: Continuation multiple instance learning for weakly supervised object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2199–2208, 2019.

[144] K. Yang, D. Li, and Y. Dou, "Towards precise end-to-end weakly supervised object detection network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8372–8381, 2019.

[145] Y. Gao, B. Liu, N. Guo, X. Ye, F. Wan, H. You, and D. Fan, "C-midn: Coupled multiple instance detection network with segmentation guidance for weakly supervised object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9834–9843, 2019.

[146] A. Arun, C. Jawahar, and M. P. Kumar, "Dissimilarity coefficient based weakly supervised object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9432–9441, 2019.

[147] Z. Zeng, B. Liu, J. Fu, H. Chao, and L. Zhang, "Wsod2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8292–8300, 2019.

[148] 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," in *NIPS Autodiff Workshop*, 2017.

[149] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[150] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1857–1866, 2018.

[151] Z. Zhang, X. Zhang, C. Peng, X. Xue, and J. Sun, "Exfuse: Enhancing feature fusion for semantic segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 269–284, 2018.

[152] 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*, pp. 4283–4292, 2020.