Semantic Scene Segmentation with Minimal Labeling Effort

Fatemehsadat Saleh

A thesis submitted for the degree of
Doctor of Philosophy at
The Australian National University

February 2020

I hereby declare that this thesis is my original work which has been done in collaboration with other researchers. This document has not been submitted to obtain a degree or award in any other university or educational institution. Parts of this thesis have been published in collaboration with other researchers in international conferences and journals as listed in the Publications section.

Fatemehsadat Saleh
18 February 2020

To my loving husband and my beloved parents

Acknowledgments

"Be grateful for whoever comes, because each has been sent as a guide from beyond . . . " –Rumi

There are a number of people without whom this thesis might not have been written, and to whom I am greatly indebted.

First and foremost, I would like to express my sincere gratitude to my supervisors, Dr. Lars Petersson, Dr. Mathieu Salzmann and Dr. Jose Alvarez for the generous support throughout my Ph.D. study, for their guidance, patience, motivation, and encouragements. I would like to thank Dr. Lars Petersson for his continuous support which were not only in the technical parts of my research but also in other situations such as funding for travel, preparing any kind of resource and hardware, or getting different kinds of access to do my research. While being professional in research, he has always been so nice and friendly so that the research environment was always enjoyable. He was always the best person to share my concerns with and then I was sure that I can trust his advice. Thank you for teaching me so much and helping me grow during my study.

I would like to express my very great appreciation to Dr. Mathieu Salzamnn who was not here, in Australia, during my Ph.D. but I have never felt this distance. During the past three years, he attended all of the remote weekly meetings from Switzerland even very early in the morning because of the time difference. His approach, vision, hard work and guidance in research enabled me to learn a lot. I will never forget his last revisions of my papers which was really precise and elegant. And I learned a lot in writing skills from him. He was one of the reasons why I decided to go to pursue a career in research. His enthusiasm and dedication for research are contagious.

I am very grateful to Dr. Jose Alvarez for his support during these years especially in the practical aspects of my research. He was the one that suggested me working on this topic, encouraged me to work on Deep Learning and introduced me the appropriate frameworks at the beginning of my study. I was really happy with this research direction during my study and I am really thankful for his suggestion. He was really dedicated and even when he moved to the US, he attended all of the weekly meetings even very late at night.

I want to thank Dr. Stephen Gould who gave detailed insightful comments about my Ph.D. research work during the first year of my Ph.D. His comments have helped me greatly in preparing my first paper which was a good starting point for my Ph.D. research.

I am also grateful to NICTA/Data61 and the Australian National University for providing my Ph.D. scholarship and enriching my academic experience by providing conference travel funding.

I thank all the present members of the Data61's Smart Vision System's Group especially Lars Andersson and Lachlan Tychsen-Smith for helping with answering questions about engineering aspects of my research especially when dealing with GPU programming, working with servers and in general making things work.

I would like to thank Sarah who is a wonderful and generous friend and will remain my best friend forever. I remember first meeting Sarah when I and my husband just arrived in Australia and she came for picking up at the airport. From the first minutes, she shared her experiences about living in Australia, working environment and the university with us. We managed to keep in touch during the year that she was studying Ph.D. in the same Lab and now we are still in touch after more than three years. I will never forget the many wonderful dinners and fun activities we have done together. Thanks so much for organizing the best surprise birthday for me. I was really lucky to have Sarah as a kind, supportive and caring friend. To my other friends, thank you for listening, offering me advice, and supporting me through this entire process. Special thanks to Mohammad E., Masoumeh, Alireza, Hajar, Mohammad N., Fatima, Alireza, and Salim. The debates, dinners, game nights, rides to the airport, and general help and friendship were all greatly appreciated.

I would like to express my gratitude to my parents-in-law for their unfailing emotional support.

A very special word of thanks goes for my inspiring parents, who have been great over the years. Your love, support and encouragement was worth more than I can express on paper. This accomplishment would not have been possible without your support. Thanks to my dear sisters Tayebe, Fahime, and Salime for their emotional support, endless love and care.

Finally, I would like to thank a very special person, my dear husband, my best friend and my amazing colleague, Sadegh for his continued and unfailing love, support and understanding during my study. I greatly value your contribution, support and suggestions to my research and deeply appreciate your belief in me. You were always around at times when I was frustrated, you celebrated with me when even the littlest things went right. I consider myself the luckiest in the world to have such a lovely and caring person, standing beside me with his love and unconditional support.

Abstract

Semantic scene segmentation – the process of assigning a semantic label to every pixel in an input image – is an important task in computer vision where an autonomous system or a robot needs to differentiate between different parts of the scene/objects and recognize the class of each one for adequate physical interactions. The most successful methods that try to solve this problem are fully-supervised approaches based on Convolutional Neural Networks (CNNs). Unfortunately, these methods require large amounts of training images with pixel-level annotations, which are expensive and time-consuming to obtain. In this thesis, we aim to alleviate the manual effort of annotating real images by designing either weakly-supervised learning strategies that can leverage image-level annotations, such as image tags, which are cheaper to obtain, or effective ways to exploit synthetic data which can be labeled automatically.

In particular, we make several contributions to the literature of semantic scene segmentation with minimal labeling effort. Firstly, we introduce a novel weakly-supervised semantic segmentation technique to address the problem of semantic scene segmentation with one of the minimal level of human supervision, image-level tags, which simply determines present and absent classes within an image. The proposed method is able to extract markedly accurate foreground/background masks from the pre-trained network itself, forgoing external objectness modules or using pixel-level/bounding box annotations, and use them as priors in an appropriate loss function. Secondly, we improve the performance of this framework by extracting class-specific foreground masks instead of a single generic foreground mask, with virtually no additional annotation cost. Thirdly, we found that a general limitation of existing tag-based semantic segmentation techniques is the assumption of having just one background class in the scene, which, by relying on the object recognition pre-trained networks or objectness modules, restricts the applicability of these methods to segmenting foreground objects only. However, in practical applications, such as autonomous navigation, it is often crucial to reason about multiple background classes. Thus, in this thesis, we introduce a weakly-supervised video semantic segmentation method in which there are multiple foreground and multiple background classes in the scene. To this end, we propose an approach to doing so by making use of classifier heatmaps. Then, we develop a two-stream deep architecture that can jointly leverage appearance and motion, and we design a new loss based on the heatmaps to train this network. In the last contribution of this thesis, we propose a novel technique for using synthetic data which lets us perform semantic segmentation without having any manual annotation, not even image-level tags. Although there exist approaches that utilize synthetic data, we use a drastically different way to handle synthetic images that does not require seeing any real images during train-

ing time. This approach builds on the observation that foreground and background classes are not affected in the same manner by the domain shift, and thus should be treated differently.

All the methods introduced in this thesis are evaluated on standard semantic segmentation datasets consisting of single background and multiple background scenes. The experiments at the end of each chapter provide compelling evidence that all of our approaches are more efficient than the contemporary baselines.

All in all, semantic scene segmentation methods with minimal labeling effort, such as those in this thesis, are crucial for having less expensive annotation processes in terms of time and money. Moreover, this will make large-scale semantic segmentation much more practical than the current models relying on full supervision, as well as lead to solutions that generalize much better than existing ones, thanks to the use of images depicting a great diversity of scenes.

Contents

List of Figures

List of Tables

Publications

The following publications have resulted from the work presented in this thesis:

- FS. Saleh, M. S. A. Akbarian, M. Salzmann, L. Petersson, S. Gould, and J. M. Alvarez. *Built-in foreground/background prior for weakly-supervised semantic segmentation*. In European Conference on Computer Vision, pages 413–432. Springer, 2016.
- FS. Saleh, M. S. A. Akbarian, M. Salzmann, L. Petersson, and J. M. Alvarez. *Bringing background into the foreground: Making all classes equal in weakly-supervised video semantic segmentation*. In: Proceedings of the IEEE International Conference on Computer Vision, pages 2106–2116, 2017.
- FS. Saleh, M. S. A. Akbarian, M. Salzmann, L. Petersson, and J. M. Alvarez, and S. Gould. *Incorporating network built-in priors in weakly-supervised semantic segmentation*. IEEE transactions on pattern analysis and machine intelligence 40, no. 6, pages 1382-1396, 2018.
- FS. Saleh, M. S. A. Akbarian, M. Salzmann, L. Petersson, and J. M. Alvarez. *Effective Use of Synthetic Data for Urban Scene Semantic Segmentation*. In European Conference on Computer Vision, pages 86-103. Springer, 2018.

Chapter 1

Introduction

Semantic scene segmentation is one of the key challenges in computer vision. In a broader view, semantic segmentation is one of the high-level tasks which makes complete scene understanding possible. The goal of scene understanding is to make machines see like humans, to have a complete understanding of visual scenes. In this context, the goal of semantic scene segmentation is to annotate each pixel of an image with a class label describing what this pixel represents. This task is also called dense prediction as the label of every pixel in the image is predicted in this process. Semantic segmentation models are useful for a variety of applications , such as autonomous driving, human-computer interaction and virtual reality (See Figure 1.1). In particular, in autonomous driving, semantic segmentation aims to equip the vehicles with the necessary perception that allows it to evolve in a constantly-changing environment. Semantic scene segmentation can provide information about free space on the road, lane markings and traffic signs, and differentiates between background classes such as road, side-walk, and grass.

1.1 Motivation

As for many other computer vision tasks, fully-supervised approaches based on Convolutional Neural Networks (CNNs) have recently achieved impressive results for semantic scene segmentation [Farabet et al., 2013; Long et al., 2015; Chen et al., 2014; Noh et al., 2015; Zheng et al., 2015; Zhang et al., 2018; Wang et al., 2018a; Lin et al., 2018]. This progress can also be seen in video semantic segmentation [Kundu et al., 2016; Tran et al., 2016; Shelhamer et al., 2016; Tripathi et al., 2015; Jin et al., 2016; Li et al., 2018; Jampani et al., 2017]. Unfortunately, these methods require large amounts of training images/videos with pixel-level annotations, which are expensive and time-consuming to obtain. For instance, pixel labeling of one Cityscapes image takes 90 minutes on average [Cordts et al., 2016]. Moreover, deploying a pre-trained semantic segmentation model to an unseen domain such as a new city whose images are not presented in the training set would not achieve satisfactory performance due to dataset biases [Chen et al., 2017b].

While semi-supervised semantic segmentation methods [Papandreou et al., 2015; Jain and Grauman, 2014a; Tsai et al., 2016a; Shankar Nagaraja et al., 2015] miti-

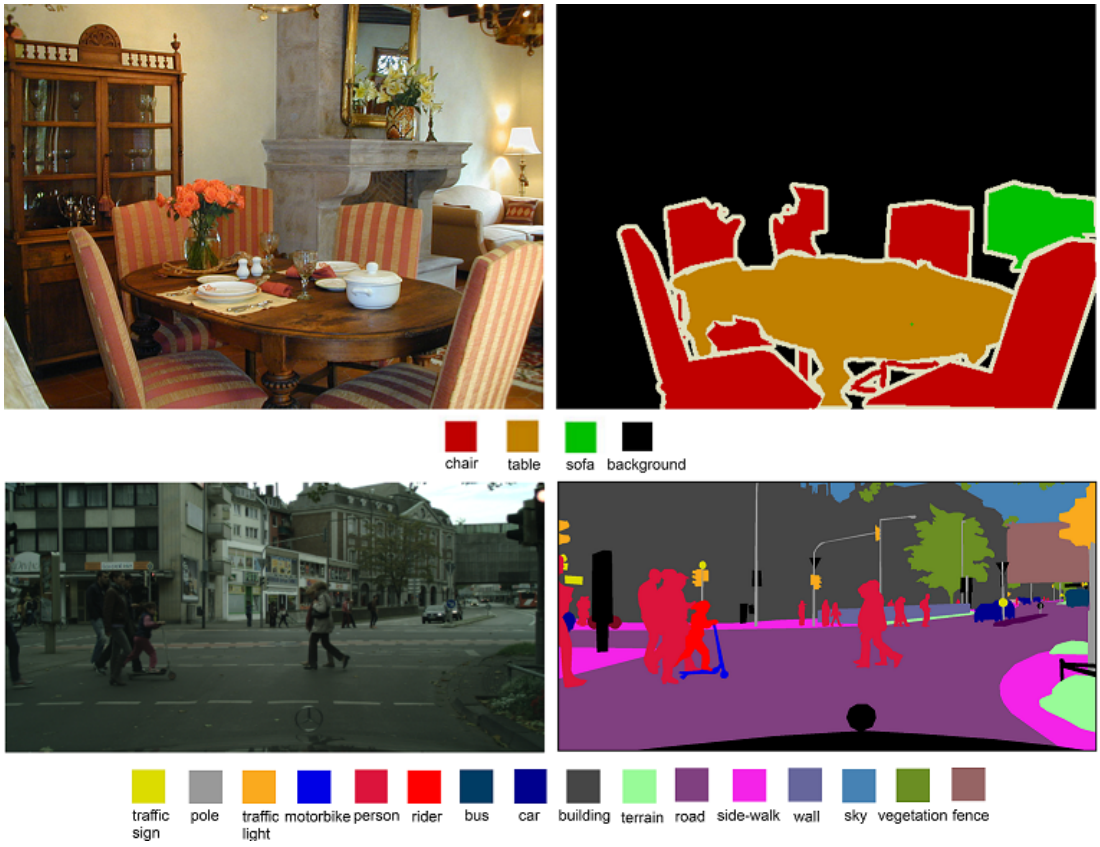


Figure 1.1: Semantic scene segmentation has many applications, such as in indoor scenes for human-robot interaction or in outdoor scenes for autonomous navigation.

gate this issue by leveraging partial annotations, they still require some pixel-level ground-truth.

Weakly-supervised semantic segmentation techniques have therefore emerged as a solution to address this limitation [Pourian et al., 2015; Xu et al., 2014; Vezhnevets et al., 2011; Xu et al., 2015; Bearman et al., 2016; Papandreou et al., 2015; Pathak et al., 2015b; Qi et al., 2015; Wei et al., 2016a,b; Kolesnikov and Lampert, 2016]. These techniques rely on a weaker form of training annotations, such as, from weaker to stronger levels of supervision, image tags [Pathak et al., 2015b; Bearman et al., 2016; Pinheiro and Collobert, 2015; Pathak et al., 2015a; Wei et al., 2016a,b; Kolesnikov and Lampert, 2016], information about object sizes [Pathak et al., 2015a], labeled points or squiggles [Bearman et al., 2016] and labeled bounding boxes [Papandreou et al., 2015; Dai et al., 2015; Khoreva et al., 2016], see Figure 1.2. In the current Deep Learning era, existing weakly-supervised methods typically start from a network pre-trained on an object recognition dataset (e.g., ImageNet [Russakovsky et al., 2015]) and fine-tune it using segmentation losses defined according to the weak annotations at hand [Pinheiro and Collobert, 2015; Pathak et al., 2015a; Papandreou et al., 2015; Bearman et al., 2016; Pathak et al., 2015b].

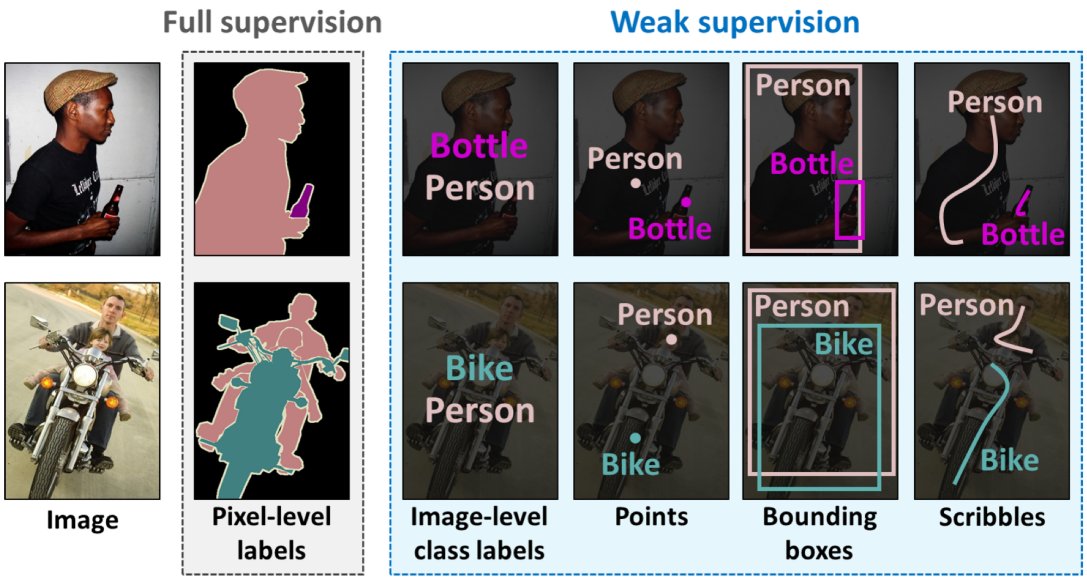


Figure 1.2: Different types of weak annotations employed for weakly supervised semantic segmentation [Hong et al., 2017a].

Among different types of weak annotations, using only image tags which are rather inexpensive attributes to annotate and thus more common in practice (e.g., Flickr [Mark J. Huiskes and Lew, 2010]) has gained increasing attention. Image tags simply determine which classes are present in the image without specifying any other information, such as the location of the objects. When working with still images [Saleh et al., 2016, 2018b; Kolesnikov and Lampert, 2016; Pathak et al., 2015a; Wei et al., 2016b; Bearman et al., 2016; Papandreou et al., 2015], tags are typically assumed to be available in each image, whereas for video-based segmentation [Hartmann et al., 2012; Liu et al., 2014; Zhang et al., 2015b; Tang et al., 2013; Wang et al., 2016; Drayer and Brox, 2016; Fragkiadaki et al., 2015; Papazoglou and Ferrari, 2013; Saleh et al., 2017], tags correspond to entire videos or video snippets. In this extreme setting, a naive weakly-supervised segmentation algorithm will typically yield poor localization accuracy. Therefore, recent works [Pinheiro and Collobert, 2015; Bearman et al., 2016; Wei et al., 2016a] have proposed to make use of objectness priors [Alexe et al., 2012; Cheng et al., 2014; Arbeláez et al., 2014; Carreira and Sminchisescu, 2010], which provide each pixel with a probability of being an object. In particular, these methods have exploited existing objectness algorithms, such as [Alexe et al., 2012; Cheng et al., 2014; Arbeláez et al., 2014], with the drawback of introducing external sources of potential error. Furthermore, [Alexe et al., 2012] typically only yields a rough foreground/background estimate, and [Cheng et al., 2014; Arbeláez et al., 2014] rely on additional training data with pixel-level annotations.

As a first contribution, in this thesis, we introduce a deep learning approach to weakly-supervised semantic segmentation where the localization information is di-

rectly extracted from networks pre-trained for the task of object recognition. Our approach relies on the following intuition: One can expect that a network trained to recognize objects in images extracts features that focus on the objects themselves, and thus has hidden layers with units firing up on foreground objects, but not on background regions. A similar intuition was also explored for object detection [Zhou et al., 2015] and localization [Oquab et al., 2015a], which inspired the weakly-supervised semantic segmentation work of [Kolesnikov and Lampert, 2016], which is contemporary to ours. In this thesis, we propose to exploit this intuition to generate (i) a foreground/background mask; and (ii) a multi-class mask which is proposed in our second contribution. We then show how these two types of masks can be incorporated in a weakly-supervised loss to train a deep network for the task of semantic segmentation using only image tags as ground-truth annotations. Ultimately, since our masks are directly extracted from pre-trained networks, our approach can be thought of as a weakly-supervised segmentation network with built-in foreground/background, or multi-class priors.

While recent years have seen great progress in weakly-supervised semantic segmentation, most existing methods, whether image- or video-based, have a major drawback: they focus on foreground object classes and treat the background as one single entity. However, having detailed information about different background classes is crucial in many practical scenarios, such as autonomous driving, where one needs to differentiate the road from a grass field. As a third contribution, in this thesis, we therefore introduce an approach to weakly-supervised video semantic segmentation that treats all classes, foreground and background, equally. To this end, we propose to rely on class-dependent heatmaps obtained from classifiers trained for image-level recognition, i.e., requiring no pixel-level annotations. These classifier heatmaps provide us with valuable information about the location of instances/regions of each class. We therefore introduce a weakly-supervised loss function that lets us exploit them in a deep architecture.

Recently, there has also been a significant effort in the community to rely on the advances of computer graphics to generate synthetic datasets [Ros et al., 2016; Richter et al., 2016, 2017; Dosovitskiy et al., 2017]. With the advance of computer graphics, generating fully-annotated synthetic data has become an attractive alternative to weakly-supervised learning. Unfortunately, despite the increasing realism of such synthetic data, there remain significant perceptual differences between synthetic and real images. Therefore, the performance of a state-of-the-art semantic segmentation network, such as [Chen et al., 2014; Long et al., 2015; Zhao et al., 2017; Noh et al., 2015], trained on synthetic data and tested on real images remains disappointingly low. While domain adaptation methods [Chen et al., 2017a; Hoffman et al., 2017, 2016; Zhang et al., 2017; Murez et al., 2017; Chen et al., 2017b] can improve such performance by explicitly accounting for the domain shift between real and synthetic data, they require having access to a large set of real images, albeit unsupervised, during training. As such, one cannot simply deploy a model trained off-line on synthetic data in a new, real-world environment. As a fourth contribution, in this thesis, we introduce a drastically different approach to addressing the mismatch between

real and synthetic data, based on the observation that not all classes suffer from the same type and degree of perceptual differences and thus should be treated differently. In particular, for the foreground classes, synthetic domain represents more accurately the shape of such classes than their texture. Thus, the foreground classes should be handled in a detection-based manner which relies more strongly on the object shape rather than on texture. Based on this observation we therefore develop a simple, yet effective semantic segmentation framework that better leverages synthetic data during training.

1.2 Contributions

The major contributions of this thesis are as follows:

1.2.1 Incorporating Network Built-in Foreground/Background Priors in Weakly-Supervised Semantic Segmentation

We propose a novel method to extract accurate foreground/background masks from the pre-trained network itself, forgoing external objectness modules. This approach relies on the following intuition: One can expect that a network trained for the task of object recognition extracts features that focus on the objects themselves, and thus has hidden layers with units firing up on foreground objects, but not on background regions. In particular, the proposed method focuses on the fourth and fifth convolution layers of the VGG16 pre-trained network, which provide higher-level information than the first three layers, such as highlighting complete objects or object parts. Then, by making use of a fully-connected Conditional Random Field (CRF), this information is smoothed out and a binary foreground/background mask is generated. Finally, the semantic segmentation results are obtained by incorporating the resulting masks as priors in the network via a weakly-supervised loss function [Saleh et al., 2016]. This approach is presented in Chapter 3.

1.2.2 Incorporating Network Built-in Multi-Class Priors in Weakly-Supervised Semantic Segmentation

We improve our previous contribution by introducing a novel method to make use of a pre-trained localization network, which specifically provides information about the location of different object classes in combination with the previous idea of using intermediate convolution layers, to obtain class-wise pixel probabilities. The final masks are obtained by making use of a fully-connected Conditional Random Field (CRF) with higher-order terms to smooth the initial pixelwise probabilities. In particular, we propose the use of the crisp boundary detection method of [Isola et al., 2014] to generate higher-order terms. We then incorporate these multi-class masks in a weakly-supervised loss function to train a Deep Network for the task of semantic segmentation using only image tags as ground-truth annotations [Saleh et al., 2018b]. This approach is presented in Chapter 3.

1.2.3 Making All Classes of Foreground and Background Equal in Weakly-supervised Video Semantic Segmentation

Most of the existing methods including our previous ones are designed to handle multiple foreground classes and a single background class. Here, we propose a novel weakly-supervised video semantic segmentation method that treats all classes, foreground and background ones, equally. To this end, we propose a method to rely on class-dependent heatmaps obtained from classifiers trained for image-level recognition, i.e., requiring no pixel-level annotations. These classifier heatmaps provide valuable information about the location of instances/regions of each class. Therefore, we introduce a weakly-supervised loss function that can exploit them in a deep architecture. In particular, we develop a two-stream deep network that jointly leverages appearance and motion. The network fuses these two complementary sources of information in two different ways: A trainable early fusion, which puts in correspondence the spatial and temporal information, and learns to combine it into a spatio-temporal stream, and a late fusion further leveraging the valuable semantic information of the spatial stream to merge it with the spatio-temporal one for final prediction [Saleh et al., 2017]. This approach is presented in Chapter 4.

1.2.4 Effective Use of Synthetic Data for Urban Scene Semantic Segmentation

Recently, automatically labeled synthetic data has been introduced for different computer vision tasks including semantic segmentation. Although these synthetic data are photo-realistic, applying a model trained on these data on a real domain will fail because of the domain shift. Here, we use synthetic data in a different way to handle this problem. Our approach builds on the observation that foreground and background classes are not affected in the same manner by the domain shift, and thus should be treated differently. In particular, the former should be handled in a detection-based manner to better account for the fact that, while their texture in synthetic images is not photo-realistic, their shape looks natural. Motivated by this fact, we propose a simple, yet effective semantic segmentation framework that better leverages synthetic data during training. In essence, the model combines the foreground masks produced by Mask R-CNN [He et al., 2017] with the pixel-wise predictions of the DeepLab semantic segmentation network [Chen et al., 2014].

Furthermore, as another contribution, we create a virtual environment in the Unity3D framework, called VEIS (Virtual Environment for Instance Segmentation). This was motivated by the fact that existing synthetic datasets [Richter et al., 2017, 2016; Ros et al., 2016] do not provide instance-level segmentation annotations for all the foreground classes of standard real datasets, such as Cityscapes. VEIS automatically annotates synthetic images with instance-level segmentation for foreground classes. It captures urban scenes using a virtual camera mounted on a virtual car. While not highly realistic, when used with a detector-based approach, this data allows us to boost semantic segmentation performance, despite it being of only little

use in a standard semantic segmentation framework [Saleh et al., 2018a]. This approach and the VEIS dataset are presented in Chapter 5.

1.3 Thesis Outline

This thesis comprises six chapters. In Chapter 2, we introduce the technical background of the methods we used in this thesis. This background material consists of some concepts and theories that are common to many of the approaches proposed in later chapters, including the convolutional neural network architectures, fully convolutional neural networks, multiple instance learning, and conditional random fields. The next three chapters propose our novel techniques for semantic scene segmentation with minimal labeling effort. In Chapter 3, we introduce our novel techniques for incorporating network built-in priors for weakly-supervised semantic segmentation. In Chapter 4, we introduce a novel approach for weakly-supervised video semantic segmentation which, in contrast to existing weakly-supervised techniques including our previous one that focus on foreground object classes and treat the background as one single entity, treats all classes, foreground and background ones, equally. In Chapter 5, we introduce a drastically different way to handle synthetic images with automatically annotated data that does not require seeing any real images at training time. Finally, in Chapter 6, we summarise the main contributions of the thesis and discuss ongoing and future work stemming from this research.

Chapter 2

Technical Background

In this chapter we introduce the background theory, architectures and models that have been used in this thesis. In this thesis, we use Convolutional Neural Networks (CNNs) widely as the learning approach. We employ the main concept of Multiple Instance Learning (MIL) to leverage weak annotations during training. In almost all of the models, we use Conditional Random Fields (CRFs) during training and also as a post-processing step in order to refine the segmentation results. Below, we review the fundamentals of CNNs, MIL, and CRFs to help the reader better understand the following chapters.

2.1 Convolutional Neural Networks

The convolutional neural networks (CNNs or ConvNets) have been established as a powerful and effective class of models to solve many problems in machine learning and computer vision, giving state-of-the-art results on image recognition [Simonyan and Zisserman, 2014; He et al., 2016], semantic segmentation [Long et al., 2015; Chen et al., 2014; Noh et al., 2015], and object detection [Girshick et al., 2014; Girshick, 2015; Ren et al., 2015]. CNNs where introduced for the first time in 1998 [LeCun et al., 1998] and the first Convolutional Neural Network was called LeNet-5 which was able to classify handwritten digits from images. CNNs are typically made of different types of layers, including convolutional, pooling, and fully-connected layers. By stacking many of these layers, CNNs can automatically learn feature representations using trainable filters and local neighborhood pooling operations which are applied to the raw input images and to the sub-sequent feature maps. This is in contrast to the traditional pattern recognition models, where a hand-designed feature extractor gathers relevant information from the input and eliminates irrelevant variabilities and then a trainable classifier categorizes the resulting feature vectors into classes.

In this section, we explain the fundamentals of CNNs including operations, learning procedure, and some important models which are mainly used in this thesis.

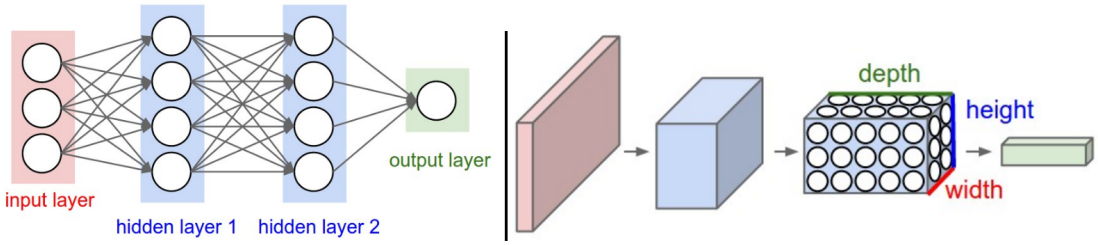


Figure 2.1: Comparison between a regular 3-layer Neural Network (left) and a CNN (right). The CNN arranges its neurons in three dimensions (width, height, and depth) and every layer of a CNN transforms the 3D input volume into a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels). [CNN, 2018]

2.1.1 Architecture Overview

Convolutional Neural Networks are a special kind of multi-layer neural networks. Like almost every other neural network, they are comprised of neurons that have learnable weights and biases, and trained with a version of the back-propagation algorithm [Rumelhart et al., 1988]. However, they have a different architecture than regular neural networks, i.e., multi-layer perceptrons.

Regular Neural Networks transform an input by putting it through a series of hidden layers which consist of a set of neurons. For regular neural networks, the most common layer type is the fully-connected layer in which neurons between two adjacent layers are fully pairwise connected. However, neurons in a single layer act completely independently and do not share any connections. Finally, there is a last fully-connected layer which is called the "output layer" and represents the class scores in a classification setting.

By contrast, convolutional neural networks take advantage of the fact that the input is an image and they constrain the architecture in a more sensible way. In particular, unlike a regular neural network, the layers of a CNN have neurons arranged in 3 dimensions: width, height, and depth. Moreover, the neurons in a layer will only be connected to a small region of the previous layer, instead of all of the neurons in a fully-connected manner. Furthermore, the final output layer will be reduced into a single vector of class scores, arranged along the depth dimension [CNN, 2018]. Figure 2.1 is a visualization of a regular neural network and a convolutional neural network.

2.1.2 Layers and Operations

A complete CNN architecture is formed by stacking different layers, with every layer transforming the activation volumes output by the previous one using a differentiable function. Figure 2.2 shows one of the classic CNN networks, AlexNet [Krizhevsky

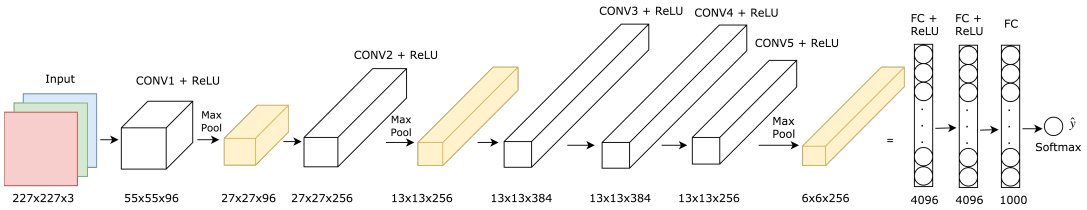


Figure 2.2: AlexNet architecture

et al., 2012] as an example of CNN architecture. Below, we discuss the most important layers and their corresponding operations.

Convolutional Layer The convolutional layer is the core building block of a CNN that does most of the computations by applying a convolution operation to the input and passing the result to the next layer. In the continuous case, the convolution of two functions f and g is defined as:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau = \int_{-\infty}^{\infty} f(t-\tau)g(\tau)d\tau \tag{2.1}$$

In the discrete case, the integral is replaced by a sum and if discrete g has support on $\{-M, ..., M\}$:

$$(f * g)(n) = \sum_{m=-M}^{M} f(n-m)g(m) \tag{2.2}$$

In this case, g is called a kernel function. All these definitions can be naturally extended to the multi-dimensional case. CNNs usually perform 2D convolution on images:

$$(f * g)(x, y) = \sum_{m=-M}^{M} \sum_{n=-N}^{N} f(x-n, y-m)g(n, m) \tag{2.3}$$

The convolutional layer's parameters consist of a set of learnable filters. Every filter is small spatially (along the width and height), but extends through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume which amounts to computing dot products between the entries of the filter and the input at any position. In fact, instead of connecting neurons to all neurons in the previous volume, each convolutional neuron will be connected to only a local region of the input volume and processes data only for its receptive field (equivalently this is the filter size). As a consequence, the network learns filters that activate when they see some certain types of visual features such as an edge in some orientation or simple color at the shallower layers, or eventually partial or entire specific patterns at higher layers of the network. Now, we have an entire set of filters in each convolutional layer, and each of them produces

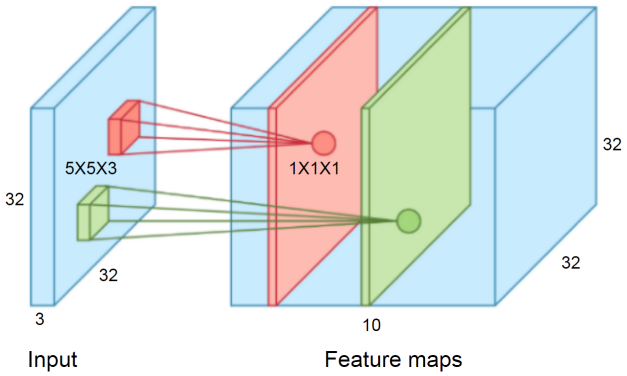


Figure 2.3: **Computing feature maps.** There are two kernels with size 5x5x3 which slide over the input and as a result two different feature map will be produced.

a separate 2-dimensional feature map. We then stack these feature maps along the depth dimension and produce the output volume. Figure 2.3 indicates how two feature maps are stacked along the depth dimension. The convolution operation for each filter is performed independently and the resulting feature maps are disjoint.

Parameter sharing is also an important concept which is used in convolutional layers to reduce the number of parameters in the whole system and makes the computation more efficient. Parameter sharing corresponds to the fact that all neurons in a particular feature map use the same filter weights. So, to obtain a particular feature map, the convolution operation is performed by sliding the filter over the input. At every location, we do an element-wise matrix multiplication and sum the result. This sum goes into the feature map.

Dilated Convolutions Dilated convolutional layers were introduced recently by [Yu and Koltun, 2015a] and rely on one more hyperparameter than the convolutional layer called the *dilation*. Applying this layer has demonstrated significant improvement especially for semantic segmentation [Chen et al., 2014]. Dilated convolutions support exponential expansion of the receptive field without loss of resolution or coverage and also without increasing the number of parameters or the amount of computation. Figure 2.4 illustrates the dilated convolution operation.

Pooling Layer Pooling is another important concept of CNNs which is a form of non-linear down-sampling and is inserted periodically in-between successive convolutional layers. Its function is to progressively reduce the spatial size of the representation. In case of Max Pooling, a spatial neighborhood (for example, a 2×2 window) is defined and the largest element from the rectified feature map within that window is taken. Instead of taking the largest element one could also take the average (Average Pooling) or sum of all elements in that window. In practice, Max Pooling is often preferred. In particular, pooling

- makes the input representations (spatial dimension) smaller and more manage-

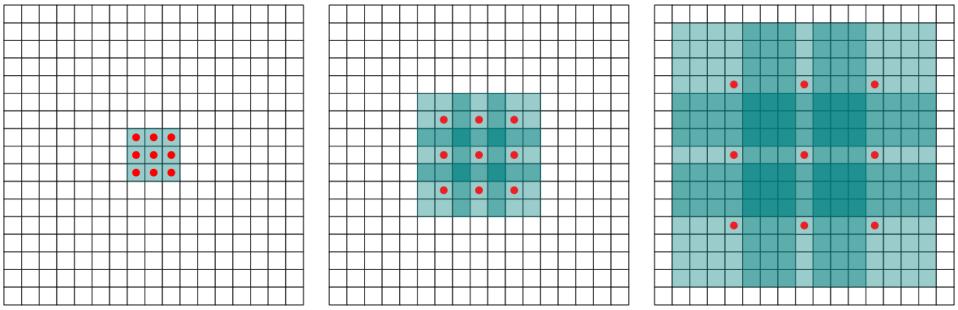


Figure 2.4: Illustration of the dilated convolutional operation which supports exponential expansion of the receptive field without loss of resolution or coverage. (a) F1 has a receptive field of 3×3. (b) F2 is produced from F1 by a 2-dilated convolution; each element in F2 has a receptive field of 7×7. (c) F3 is produced from F2 by a 4-dilated convolution; each element in F3 has a receptive field of 15×15. The number of parameters associated with each layer is identical [Yu and Koltun, 2015a].

able

- reduces the number of parameters and computations in the network, therefore, controlling overfitting
- makes the network invariant to small transformations, distortions and translations in the input image as a small distortion in the input will not change the output of pooling since we take the maximum/average value in a local neighborhood
- helps to have an almost scale invariant representation of the image

ReLU Activation ReLU stands for Rectified Linear Unit and is a non-linear and element-wise operation. Its output is given by:

$$f(x) = max(0, x) \tag{2.4}$$

For any kind of neural network to be expressive, it needs to contain non-linearity. The purpose of ReLU is to introduce non-linearities in the CNNs [Glorot et al., 2011], since most of the real-world data a CNNs should learn would be non-linear. Therefore, the result of the convolution operation is passed through a ReLU activation function. Other non-linear functions, such as tanh or sigmoid, can also be used instead of ReLU, but ReLU has been found to perform better in most situations.

Fully Connected Layer In fully connected layers, neurons are connected to all activations in the previous layer which in principle is the same as the traditional regular neural network. In a CNN, a number of fully connected layers are added to complete

the architecture. In fact, the output from the convolutional and pooling layers represent high-level features of the input and the purpose of the fully connected layer is to use these features for classifying the input into various classes based on the training dataset. Moreover, adding a fully-connected layer is a way of learning non-linear combinations of these features. A fully connected layer expects a 1D real-valued vector as input. Therefore, the output of the final pooling layer, is a 3D volume, flattened to a vector. This vector becomes the input to the fully connected layer which produces non-spatial outputs. A Softmax function is usually used as the activation function in the output layer of the fully connected layer which takes a vector of arbitrary real-valued scores and converts it to a vector of values between zero and one that sum to one, i.e, in the form of a probability.

2.1.3 Learning

As we discussed in the previous sections, a CNN is basically a combination of two components: the feature extraction part and the classification part. The convolution and pooling layers perform feature extraction, and the fully connected layers then act as a classifier on top of these features and assign a vector of scores to each input. In this way, CNNs transform the original input (image) layer by layer from the original pixel values to the final class scores. Note that some layers contain parameters and others do not. In particular, the convolutional and fully connected layers perform transformations that are a function of activations in the input volume and the parameters (the weights and biases of the neurons). On the other hand, the ReLU/Pooling layers implement a non-parametric function.

The network parameters in the convolutional and fully connected layers are trained using the back-propagation [Rumelhart et al., 1988] as in the case of the regular neural networks. Back-propagation, short for "backward propagation of errors," is an algorithm for supervised learning of neural networks using gradient descent. Given a neural network and an error function (loss function), the method calculates the gradient of the error function with respect to the neural network's parameters.

Consider a dataset consisting of input-output pairs (x_i, y_i) where x_i is the input and y_i is its corresponding desired output of the network or label. Let us denote by $X = \{(x_1, y_1), ..., (x_N, y_N)\}$, the set of pairs of size N. The parameters of a neural network are denoted as θ which contains, for each node i weights w_i and bias b_i. There is also, an objective function $f(X, \theta)$ (sometimes referred to as the cost function or loss function) which calculates the difference between the network output $\hat{y}_i$ and its expected output y_i for a set of input-output pairs $(x_i, y_i) \in X$ and a particular value of θ. Training a neural network using gradient descent requires the calculation of the gradient of the objective function $f(X, \theta)$ with respect to the weights and biases (collectively denoted θ). Then, according to the learning rate α, which is the weight of the negative gradient, each iteration of gradient descent updates the network parameters according to:

$$\theta^{t+1} = \theta^t - \alpha \frac{\partial f(X, \theta^t)}{\partial \theta} \tag{2.5}$$

where θ^t denotes the parameters of the neural network at iteration t of the gradient descent.

The standard gradient descent algorithm may be infeasible when the training data size is huge. However, Stochastic Gradient Descent (SGD), also known as incremental gradient descent, is an iterative method for optimizing a differentiable loss function [Robbins and Monro, 1985] which has received a considerable amount of attention just recently in the context of large-scale learning. When training deep learning models, the objective function is often considered as a sum of a finite number of functions:

$$f(X) = \frac{1}{N} \sum_{i=1}^{n} f_i(X) \tag{2.6}$$

where $f_i(x)$ is a loss function based on the training data instance indexed by i. Therefore, for a standard gradient descent approach, the computational cost in each iteration scales linearly with the training data set size N. Stochastic gradient descent offers an efficient solution. At each iteration, rather than computing the gradient $\nabla f(x)$, SGD randomly samples i uniformly and computes $\nabla f_i(x)$ instead. In other words, SGD uses $\nabla f_i(x)$ as an unbiased estimator of $\nabla f(x)$ since:

$$\mathbb{E}_i \nabla f_i(x) = \frac{1}{N} \sum_{i=1}^{N} \nabla f_i(x) = \nabla f(x) \tag{2.7}$$

In a generalized case, at each iteration a mini-batch that consists of indices for training data instances may be sampled uniformly with replacement. In fact, stochastic gradient descent typically reaches convergence much faster than standard gradient descent since it updates weight more frequently. In this thesis, SGD is used as the optimization method during the CNN training.

2.1.4 Fully Convolutional Neural Network (FCN)

In order to have a CNN architecture for semantic segmentation, one approach is to adapt classifier networks for dense prediction. Typical recognition networks such as VGG [Simonyan and Zisserman, 2014], ResNet [He et al., 2016], AlexNet [Krizhevsky et al., 2012], take fixed-sized inputs and produce non-spatial outputs. As mentioned before, the fully connected layers of these networks have fixed dimensions and throw away spatial coordinates. However, these fully connected layers can also be viewed as convolutions with kernels that cover their entire input regions and convert these networks to "Fully Convolutional Networks".

The basic idea behind a fully convolutional network is that it is "fully convolutional", that is, all of its layers are convolutional layers. FCNs do not have any fully-connected layers at the end, which are typically used for classification. Instead, they use convolutional layers which aim to learn representations and make decisions based on local spatial input. The only difference needed in the architecture of a classifier network to be changed is to convert fully connected layers into convolutional layers with the right filter size [Long et al., 2015]. This transformation is illustrated

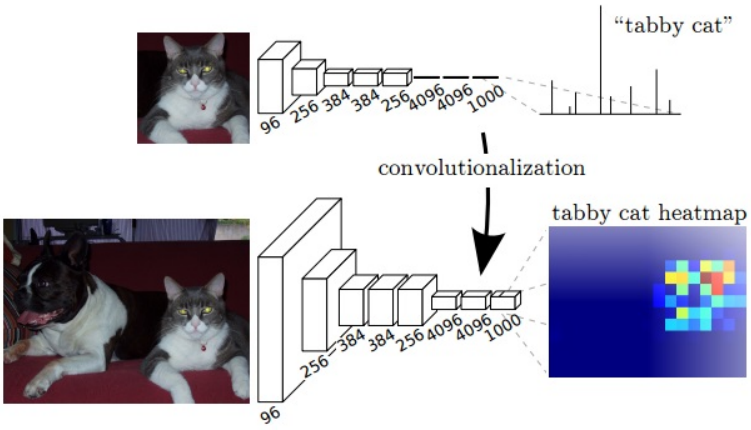


Figure 2.5: Transforming fully connected layers into convolution layers enables a classification network to output a heatmap. Adding layers and a spatial loss produces an efficient machine for end-to-end dense learning [Long et al., 2015].

in Figure 2.5.

To obtain a segmentation map (output), segmentation networks usually have 2 parts: A downsampling path, which is used to capture semantic/contextual information, and an upsampling path, which is used to recover spatial information. In order to fully recover the fine-grained spatial information lost in the pooling or downsampling layers, there are different approaches. The authors of [Long et al., 2015] use bilinear interpolation which computes each output y_{ij} from the nearest four inputs by a linear map that depends only on the relative positions of the input and output cells. In a sense, upsampling with a factor of f is a convolution with a fractional input stride of $1/f$. So long as f is integral, a natural way to upsample is therefore backwards convolution (sometimes called deconvolution) with an output stride of f. Thus upsampling is performed in-network for end-to-end learning by backpropagation from the pixelwise loss. Note that the deconvolution filter in such a layer need not be fixed (e.g., to bilinear upsampling), but can be learned. A stack of deconvolution layers and activation functions can even learn a nonlinear upsampling. The authors of [Chen et al., 2014] skip subsampling after the last two max-pooling layers in the VGG network architecture and modify the convolutional filters in the layers that follow them by introducing zeros to increase their length. This allows them to compute dense CNN feature maps at any target subsampling rate without introducing any approximations. The module introduced in [Yu and Koltun, 2015a] uses dilated convolutions to systematically aggregate multi-scale contextual information without losing resolution.

2.2 Multiple Instance Learning

One of the drawbacks of the supervised learning model is that it is not always possible for a teacher to provide labeled examples for training. Multiple-instance learning

(MIL), which is a form of weakly-supervised learning, provides a new way of modeling the teacher's weaknesses. Instead of receiving a set of instances which are labeled positive or negative, the learner receives a set of bags that are labeled positive or negative. Each bag contains many instances. A bag is labeled negative if all the instances in it are negative. On the other hand, a bag is labeled positive if there is at least one instance in it which is positive. From a collection of labeled bags, the model tries to learn how to label individual instances correctly. This problem is even harder than noisy supervised learning since the ratio of negative to positive instances in a positively-labeled bag can be arbitrarily high [Maron and Lozano-Pérez, 1998; Carbonneau et al., 2018].

2.2.1 MIL For Semantic Segmentation

[Pathak et al., 2015b] proposed an effective MIL formulation of multi-class semantic segmentation learning using a fully convolutional network. This model and the proposed loss function is an important baseline used in Chapters 3 and 4 of this thesis. In this method, a model is trained for a semantic segmentation from just image tags. The model is a fully convolutional network which accepts inputs of any size and produces a pixel-wise score map for each class. The proposed multi-class MIL loss further exploits the supervision given by images with multiple labels. To learn the segmentation model from image tags, each image is considered as a bag of pixel-level-instances with a pixel-wise, multi-class form of MIL loss.

A multi-class MIL loss is defined as the multi-class logistic loss computed at maximum predictions, as discussed below. For an image of any size, the FCN outputs a heatmap for each class (including the background) of the corresponding size. The max scoring pixel is identified in the coarse heatmaps of the classes present in the image and the background. The loss is then only computed on these coarse points, and is back propagated through the network. The background class is similar to the negative instances and competes against the positive object classes. Let the input image be I, its label set be $\mathcal{L}$ (including the background label) and $\hat{p}_l(x, y)$ be the output heatmap for the l^{th} label at location (x, y). The loss is defined as:

$$(x_l, y_l) = argmax_{x,y} \hat{p}_l(x, y) \qquad \forall l \in \mathcal{L}_I \tag{2.8}$$

$$Loss = \frac{-1}{|\mathcal{L}_I|} \sum_{l \in \mathcal{L}_I} log \hat{p}_l(x_l, y_l) \tag{2.9}$$

During inference, the MIL-FCN takes the top class prediction at every point in the coarse prediction and by using bilinear interpolation produces a pixel-wise segmentation of the input image resolution.

2.3 Conditional Random Fields

Conditional Random Fields (CRFs) [Lafferty et al., 2001], a type of discriminative probabilistic graphical model, are generally used to segment and label sequence data. In this section, we provide a brief overview of CRFs for pixel-wise labeling which is the main topic in this thesis. A CRF, used in the context of pixel-wise label prediction, models pixel labels as random variables that form a Markov Random Field (MRF) when conditioned upon a global observation. The global observation is usually taken to be the image. Let X_i be the random variable associated to pixel i, which represents the label assigned to pixel i and can take any value from a predefined set of labels $\mathcal{L} = \{l_1, l_2, ..., l_{\mathcal{L}}\}$. Let X be the vector formed by the random variables $X_1, X_2, ..., X_N$, where N is the number of pixels in the image. Given a graph $G = (V, E)$, where $V = \{X_1, X_2, ..., X_N\}$, and a global observation (image) I, the pair (I, X) can be modelled as a CRF characterized by a Gibbs distribution of the form $P(X = x|I) = \frac{1}{Z(I)} exp(-E(x|I))$. Here, $E(x)$ is called the energy of the configuration $x \in \mathcal{L}^N$ and $Z(I)$ is the partition function. [Krähenbühl and Koltun, 2011] introduced a specific type of CRF, which has been named fully connected pairwise CRF or dense CRF. This model is defined on the complete set of pixels in an image. The resulting graphs have billions of edges, making traditional inference algorithms impractical. However, in dense CRF an efficient approximation inference algorithm is proposed in which the pairwise edge potentials are defined by a linear combination of Gaussian kernels. The energy of a label assignment x is given by:

$$E(x) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j) \tag{2.10}$$

where the unary energy components $\psi_u(x_i)$ measure the inverse likelihood (the cost) of pixel i taking label x_i, and the pairwise energy components $\psi_p(x_i, x_j)$ measure the cost of assigning labels x_i, x_j to pixels i, j simultaneously. The pairwise energies provide an image data-dependant smoothing term that encourages assigning similar labels to pixels with similar properties. The pairwise potentials are modelled as weighted Gaussians:

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{M} w^{(m)} k_G^{(m)}(f_i, f_j) \tag{2.11}$$

where each $k_G^{(m)}$, for $m = 1, ..., M$, is a Gaussian kernel applied to the feature vectors. The feature vector of pixel i, denoted by f_i, is derived from image features such as spatial location and RGB values. The function $\mu(.,.)$ called the label compatibility function, captures the compatibility between different pairs of labels as the name implies.

Minimizing the above CRF energy $E(x)$ yields the most probable label assignment x for the given image. Since this exact minimization is intractable, a mean-field approximation to the CRF distribution is used for approximate maximum posterior

marginal inference. It consists of approximating the CRF distribution $P(X)$ by a simpler distribution $Q(X)$, which can be written as the product of independent marginal distributions, i.e, $Q(X) = \prod_i Q_i(X_i)$. Each iteration of the iterative algorithm for approximate mean-field inference performs a message passing step, a compatibility transform, and a local update. Both the compatibility transform and the local update run in linear time and are highly efficient. The computational bottleneck is message passing. For each variable, this step requires evaluating a sum over all other variables. A naive implementation thus has quadratic complexity in the number of variables N. However, [Krähenbühl and Koltun, 2011] performed the message passing using Gaussian filtering in feature space. This enables utilizing highly efficient approximations for high-dimensional filtering, which reduce the complexity of message passing from quadratic to linear.

2.4 Summary

In summary, in this chapter, we have covered the technical backgrounds that have been used in the following chapters. In particular, we presented the building blocks of a CNN architecture which has been widely used as the learning approach in this thesis. We also reviewed the main concepts in MIL for weakly-supervised learning. Moreover, we briefly presented the concept of CRFs which have been used as a post-processing step as well as in learning in the approaches proposed in this thesis.

Chapter 3

Incorporating Network Built-in Priors in Weakly-Supervised Semantic Segmentation

Semantic scene segmentation, i.e., assigning a class label to every pixel in an input image, has received growing attention in the computer vision community, with accuracy greatly increasing over the years. Despite the success of fully-supervised approaches, these methods require large amounts of training images with pixel-level annotations, which are expensive and time-consuming to obtain. Weakly-supervised techniques, which rely on a weaker form of training annotations, have therefore emerged as a solution to address this limitation. In particular, weak supervision using only image tags has gained much attention due to its cheaper annotation cost. To this end, CNN-based methods have been proposed to fine-tune pre-trained networks using image tags. However, without additional information, this leads to poor localization accuracy. This problem, however, was alleviated by making use of objectness priors to generate foreground/background masks. Unfortunately, these priors require pixel-level annotations/bounding boxes, or will still yield inaccurate object boundaries. In this chapter, we propose a novel method to extract accurate masks from networks pre-trained for the task of object recognition, thus forgoing external objectness modules. We first show how foreground/background masks can be obtained from the activations of higher-level convolutional layers of a network. We then show how to obtain multi-class masks by the fusion of foreground/background masks with information extracted from a weakly-supervised localization network. Our experiments evidence that exploiting these masks in conjunction with a weakly-supervised training loss yields state-of-the-art tag-based weakly-supervised semantic segmentation results.

3.1 Introduction

In this chapter, we are particularly interested in exploiting one of the weakest levels of supervision, i.e., image tags, which has gained much attention in recent years. In

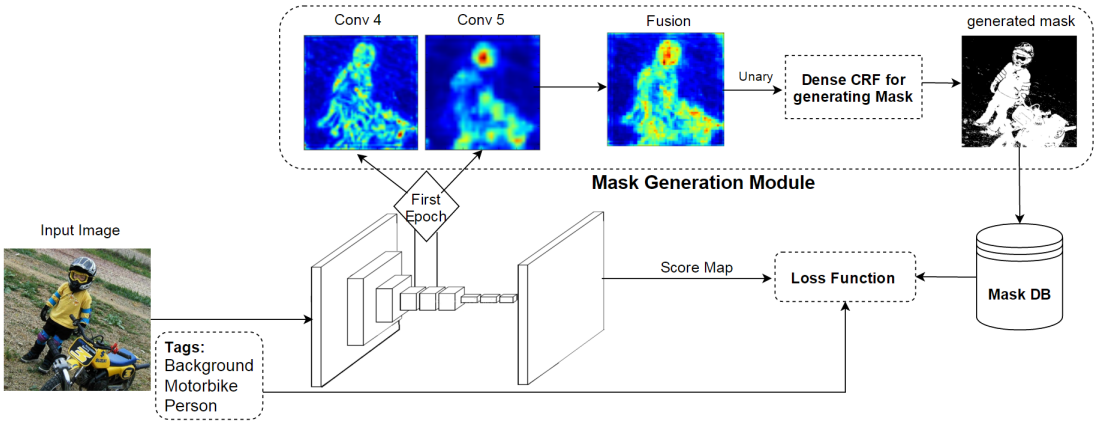


Figure 3.1: Overview of our weakly-supervised network with built-in foreground/background prior.

contrast to most of the existing methods which use additional objectness or localization information to gain better segmentation results [Pinheiro and Collobert, 2015; Bearman et al., 2016; Wei et al., 2016a], we are interested in obtaining the objectness cues from the network itself without using additional annotations for training a localization or objectness module.

More specifically, starting from a fully-convolutional network, pre-trained on ImageNet, we propose to extract a foreground/background mask by directly exploiting the unit activations of some of the hidden layers in the network. In particular, as illustrated in Figure 3.1, we focus on the fourth and fifth convolution layers of the VGG-16 pre-trained network [Simonyan and Zisserman, 2014], which provide higher-level information than the first three layers, such as highlighting complete objects or object parts. Note that the resulting masks can also be thought of as a form of objectness measure. While several CNN-based approaches have proposed to learn objectness, or saliency measures from annotations Ghodrati et al. [2015]; Kuo et al. [2015]; Zou and Komodakis [2015], to the best of our knowledge, our approach is the first to extract this information directly from the hidden layer activations of a segmentation network, and employ the resulting masks as localization cues for weakly-supervised semantic segmentation. Furthermore, we extend our framework to incorporate some additional, yet cheap, supervision, taking the form of asking the user to select the best foreground/background mask among several automatically generated candidates. Our experiments reveal that this additional supervision only costs the user roughly 2-3 seconds per image and yields another significant accuracy boost over our tags-only results. While effective, this approach only reasons about foreground/background, without explicitly considering the different foreground classes. To address this, we propose to make use of a pre-trained localization network, which specifically provides information about the location of different object classes. We then show how this information can be combined with the previous fusion-based strategy, as illustrated in Figure 3.2, to obtain class-wise pixel probabilities. In both

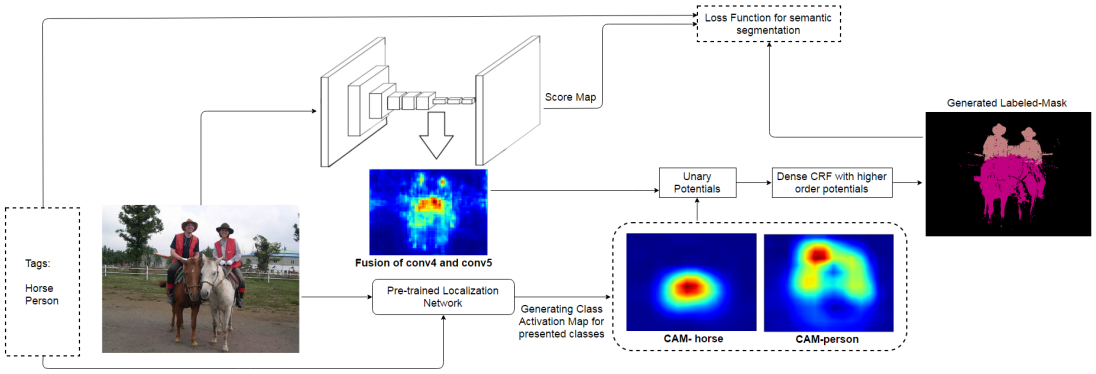


Figure 3.2: Overview of our weakly-supervised network with multi-class masks.

the foreground/background and multi-class cases, the final masks are obtained by making use of a fully-connected Conditional Random Fields (CRF) with higher-order terms to smooth the initial pixel-wise probabilities. In particular, we propose to make use of the crisp boundary detection method of [Isola et al., 2014] to generate our higher-order terms.

We then show how these two types of masks can be incorporated in a weakly-supervised loss to train a Deep Network for the task of semantic segmentation using only image tags as ground-truth annotations. Ultimately, since our masks are directly extracted from pre-trained networks, our approach can be thought of as a weakly-supervised segmentation network with built-in foreground/background, or multi-class prior.

We demonstrate the benefits of our approach on Pascal VOC 2012 [Everingham et al., 2015], which is the most popular dataset for weakly-supervised semantic segmentation. Our experiments show that our approach outperforms the state-of-the-art methods that use image tags only, and even some methods that leverage additional supervision, such as object size information [Pathak et al., 2015a] and point supervision [Bearman et al., 2016]. To demonstrate the generality of our approach, we also report results on two other challenging datasets: YouTube Objects [Prest et al., 2012] and Microsoft COCO [Lin et al., 2014]. To the best of our knowledge, this represents the first attempt at performing weakly-supervised semantic segmentation on MS-COCO.

In this chapter, first we focus on foreground/background masks and then we introduce an approach to generating class-specific masks and employing them for weakly-supervised semantic segmentation. Furthermore, we introduce new higher-order terms in our CRF by exploiting the crisp boundary detection framework [Isola et al., 2014]. Finally, in addition to producing state-of-the-art results, our experiments provide a thorough evaluation of the different components of our model.

3.2 Related Work

Weakly-supervised semantic segmentation has attracted a lot of attention, because it alleviates the painstaking process of manually generating pixel-level training annotations. Over the years, great progress has been made [Xu et al., 2014; Vezhnevets et al., 2011; Xu et al., 2015; Bearman et al., 2016; Papandreou et al., 2015; Pathak et al., 2015b; Pinheiro and Collobert, 2015; Pathak et al., 2015a; Dai et al., 2015; Vezhnevets et al., 2012; Wei et al., 2016a,b; Kolesnikov and Lampert, 2016; Qi et al., 2016; Shimoda and Yanai, 2016]. In particular, recently, Convolutional Neural Networks (CNNs) have been applied to the task of weakly-supervised segmentation with great success. In this section, we discuss these CNN-based approaches, which are the ones most related to our work.

The work of [Pathak et al., 2015b] constitutes the first method to consider fine-tuning a CNN pre-trained for object recognition, using image tags only, within a weakly-supervised segmentation context. This approach relies on a Multiple Instance Learning (MIL) loss to account for image tags during training. While this loss improves segmentation accuracy over a naive baseline, this accuracy remains relatively low, due to the fact that no other prior than image tags is employed. By contrast, [Papandreou et al., 2015] incorporates an additional prior in the MIL framework in the form of an adaptive foreground/background bias. This bias significantly increases accuracy, which [Papandreou et al., 2015] shows can be further improved by introducing stronger supervision, such as labeled bounding boxes. Importantly, however, this bias is data-dependent and not trivial to re-compute for a new dataset. Furthermore, the results remain inaccurate in terms of object localization. In [Pathak et al., 2015a], weakly-supervised segmentation is formulated as a constrained optimization problem, and an additional prior modeling the size of objects is introduced. This prior relies on thresholds determining the percentage of the image area that certain classes of objects can occupy, which again is problem-dependent. More importantly, and as in [Papandreou et al., 2015], the resulting method does not exploit any information about the location of objects, and thus yields poor localization accuracy.

To overcome this weakness, some approaches [Pinheiro and Collobert, 2015; Bearman et al., 2016; Wei et al., 2016a; Qi et al., 2016] have proposed to exploit the notion of objectness. In particular, [Pinheiro and Collobert, 2015] makes use of a post-processing step that smoothes initial segmentation results using the object proposals obtained by BING [Cheng et al., 2014] or MCG [Arbeláez et al., 2014]. While it improves localization, being a post-processing step, this procedure is unable to recover from some mistakes made by the initial segmentation. By contrast, [Bearman et al., 2016; Wei et al., 2016a] directly incorporate an objectness score [Alexe et al., 2012; Arbeláez et al., 2014] in their loss function. [Qi et al., 2016] also uses these objectness methods to generate segmentation masks and train the semantic segmentation network iteratively. While accounting for objectness when training the network indeed improves segmentation accuracy, the whole framework depends on the success of the external objectness module, which, in practice, only produces a coarse heatmap and does not accurately determine the location and shape of the objects (as evidenced by

our experiments). Note that BING and MCG have been trained from PASCAL *train* images with full pixel-level annotations or bounding boxes, and thus [Pinheiro and Collobert, 2015; Wei et al., 2016a; Qi et al., 2016] inherently make use of stronger supervision than our approach. Instead of objectness, the method in [Wei et al., 2016b] relies on DRIF saliency maps [Jiang et al., 2013]. These saliency maps are employed to train a simple network from Flickr images, whose output then serves to train two other networks using more complicated Pascal VOC images. Note that, again, the DRIF method requires bounding boxes in its training stage, thus inherently making use of additional supervision. [Shimoda and Yanai, 2016] tried to produce class-specific saliency maps based on the derivatives of the class scores w.r.t. the input image that provides some localization cues for segmentation. The method of [Tokmakov et al., 2016] also uses motion cues of weakly annotated videos to segment images with a subset of the PASCAL VOC classes. Here, instead of relying on an external objectness or saliency method, we leverage the intuition that, within its hidden layers, a network pre-trained for object recognition should already have learned to focus on the objects themselves. This lets us generate a foreground/background mask directly from the information built into the network, which we empirically show provides a more accurate object localization prior.

Beyond foreground/background masks, the method of the contemporary work [Kolesnikov and Lampert, 2016] exploits the output of the same localization network [Zhou et al., 2016] as us, but directly in a new composite loss function for weakly-supervised semantic segmentation. While effective, the method suffers from the fact that localization of some classes is inaccurate. By contrast, here, we combine our built-in foreground/background mask with information from the localization network, thus obtaining more accurate multi-class masks. As evidenced by our experiments, these more robust masks yield more accurate semantic segmentation results.

There are also some very recent approaches that use image tags for semantic segmentation. Most of these approaches try to expand the object regions from discriminative parts to non-discriminative parts to cover the whole objects and then use these dense localization maps for semantic segmentation. [Wei et al., 2018] uses a classification network equipped with convolutional blocks of different dilated rates which can enlarge the receptive fields of convolutional kernels and transfer the surrounding discriminative information to non-discriminative object regions and utilize these regions in the object localization maps which would be beneficial for weakly-supervised semantic segmentation. [Wei et al., 2017] is another method which uses adversarial erasing to iteratively train multiple classification networks for expanding discriminative regions. [Huang et al., 2018] proposes a semantic segmentation network starting from the discriminative regions and progressively increase the pixel-level supervision using seeded region growing. [Wang et al., 2018b] uses an iterative bottom-up and top-down framework which, starting from coarse but discriminative object seeds, mine common object features from them to expand object regions and then uses a saliency-guided refinement method to supplement non-discriminative parts. Then, in the top-down step, these regions are used as supervision to train

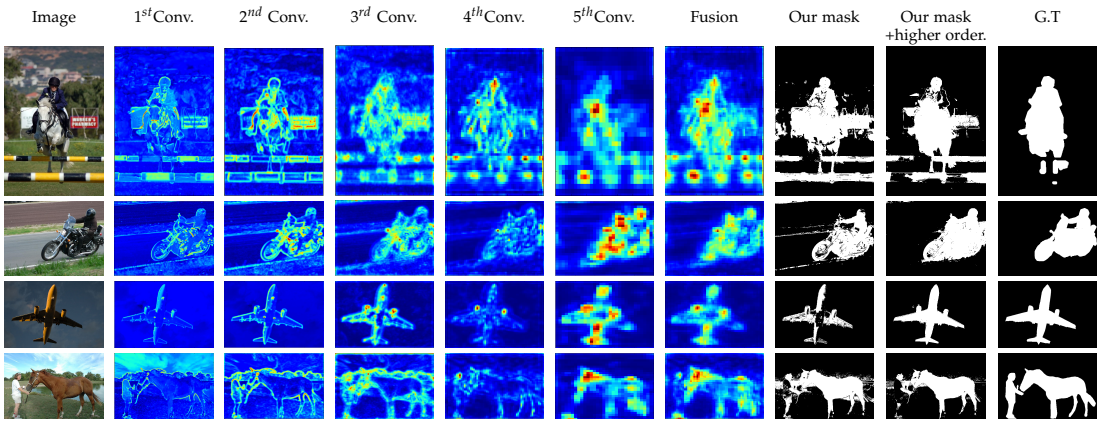


Figure 3.3: **Built-in foreground/background mask.** From left to right, we show the input image, the activations of the first, second, third, fourth, and fifth convolutional layers, the results of our fusion strategy, and the final mask after CRF smoothing without and with higher order term followed by the ground-truth mask. Note that "Fusion" constitutes the unary potential of the dense CRF used to obtain "Our mask".

the segmentation network and predict segmentation masks. Another promising way to improve the segmentation performance is to utilize additional weakly supervised images, such as web images, to train CNNs [Jin et al., 2017]. There are also some methods which use saliency models with additional supervision such as bounding boxes. As an example [Oh et al., 2017] employs saliency which is trained from bounding box annotations as additional information and hereby exploit prior knowledge on the object extent and image statistics for object segmentation from image tags.

3.3 Our Method

In this section, we introduce our weakly-supervised semantic segmentation framework. First, we present our approach to extracting masks, either foreground/background or multi-class, directly from a network pre-trained for object recognition. We then introduce our weakly-supervised learning algorithms that leverage these foreground/background and multi-class masks.

3.3.1 Built-in Prior Models

Given an image, our goal is to automatically extract a mask that indicates which regions correspond to either foreground/background or specific classes. The central idea of our approach is to rely on networks that have been pre-trained for object recognition. Intuitively, we expect that such networks have learned to focus on the objects themselves, and their parts, rather than on background regions. Below, we show how we can exploit this intuition to extract foreground/background masks, as well as multi-class ones.

3.3.1.1 Foreground/Background Masks

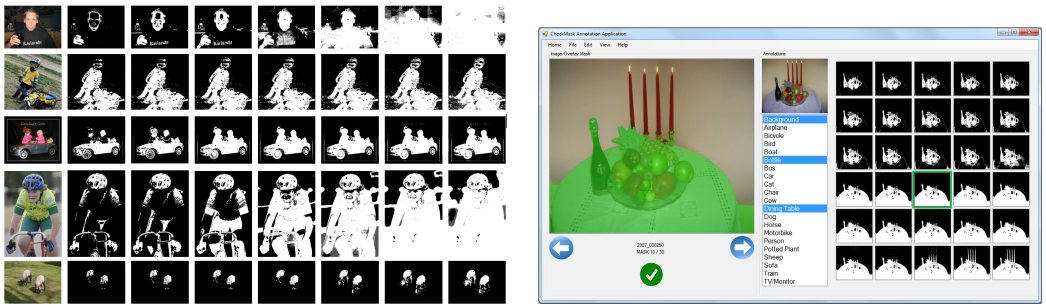
Let us first consider the case of foreground/background masks. In practice, as discussed in more detail in Section 3.4.2.1, we make use of an architecture based on the VGG-16 network [Simonyan and Zisserman, 2014], whose weights were trained on ImageNet [Russakovsky et al., 2015] for the task of object recognition, converted into a fully-convolutional network. If, to recognize objects, the network has learned to focus on the objects themselves, it should produce high activation values on the objects and on their parts. To evaluate this, we studied the activations of the different hidden layers of our initial network.

More specifically, we passed each image forward through the network and visualized each activation by computing the mean over the channels after resizing the activation map to the input image size. Perhaps unsurprisingly, this lead to the following observations, illustrated in Figure 3.3. The first two convolutional layers of the VGG network extract image edges. As we move deeper in the network, the convolutional layers extract higher-level features. In particular, the third convolutional layer fires up on prototypical object shapes. The fourth and fifth layers indicate the location of complete objects and of their most discriminative parts. Note that a similar study was performed in the different context of edge detection [Bertasius et al., 2014], with similar conclusions.

Based on these observations, we propose to make use of the fourth and fifth layers to produce an initial foreground/background mask estimate. To this end, we first convert these two layers from 3D tensors ($512 \times W \times H$) to 2D matrices ($W \times H$) via an average pooling operation over the 512 channels. We then fuse the two resulting matrices by simple elementwise summation, and scale the resulting values between 0 and 1. The resulting $W \times H$ map can be thought of as a pixel-wise foreground probability, which we denote by P_f in the remainder of the chapter. Figure 3.3 illustrates the results of this method on a few images from PASCAL VOC 2012. Note that, while the resulting scores indeed accurately indicate the location of the foreground objects, this initial mask remains noisy. This will be addressed in Section 3.3.1.3 by encouraging smoothness via a CRF.

Our foreground/background masks can be thought of as a form of objectness measure. While objectness has been used previously for weakly-supervised semantic segmentation (MCG and BING in [Pinheiro and Collobert, 2015; Qi et al., 2016; Wei et al., 2016a], and the generic objectness [Alexe et al., 2012] in [Bearman et al., 2016]), the benefits of our approach are twofold. First, we extract this information directly from the same network that will be used for semantic segmentation, which prevents us from having to rely on an external method. Second, as opposed to BING and MCG, we require neither object bounding boxes, nor object segments to train our method. Finally, as shown in our experiments, our method yields much more accurate object localization than the techniques in [Alexe et al., 2012] and [Arbeláez et al., 2014], which typically only provide a rough outline of the objects.

However, the masks obtained with the proposed approach are not always perfect. This is due to the fact that the information obtained by fusing the activations of the



(a) (b)

Figure 3.4: (a) Mask candidates generated with our approach. From left to right, we show the input image, the 1^{st}, 5^{th}, 10^{th}, 15^{th}, 20^{th}, 25^{th} and 30^{th} solutions. (b) Our new level of supervision: The annotator selects a mask which he/she thinks contains all foreground object(s) and the minimum amount of background.

fourth and fifth layers is noisy, and thus the solution found by inference in the CRF is not always the desired one. As a matter of fact, many other solutions also have a low energy (Equation 3.6). Rather than relying on a single mask prediction, we propose to generate multiple such predictions, and provide them to a user who decides which one is the best one.

The problem of generating several predictions in a given CRF is known as the *M*-best problem. Here, in particular, we are interested in generating solutions that all have low energy, but are diverse, and thus follow the approach of [Batra et al., 2012]. In essence, this approach iteratively generates solutions, and, at each iteration, modifies the energy of Equation 3.6 to encourage the next solution to be different from the ones generated previously. In practice, we make use of the Hamming distance as a diversity measure. This diversity measure can be encoded as an additional unary potential in Equation 3.6, and thus comes at virtually no additional cost in the inference procedure. For more details about the diverse *M*-best strategy, we refer the reader to [Batra et al., 2012].

Ultimately, we generate several masks with this procedure, and ask the user to click on the one that best matches the input image. Such a selection can be achieved very quickly. In practice, we found that a user takes roughly 2-3 seconds per image to select the best mask. As a consequence, this new source of weak supervision remains very cheap, while, as evidenced by our experiments, allows us to achieve a significant improvement over our tags-only formulation (Figure 3.4).

3.3.1.2 Multi-class Masks

The main drawback of the foreground/background masks discussed above is that they are not class-specific. The network we used to extract these masks has not been fine-tuned with the desired classes, and thus the activations only provide information about the location of generic foreground objects. Here, we address this limitation by making use of a class-specific localization network [Zhou et al., 2016] in conjunction

with our foreground/background masks.

The main idea behind the localization network of [Zhou et al., 2016] is to generate a Class Activation Map (CAM) for each specific object category, or, in other words, a heatmap indicating the location of the regions that are useful for the network to recognize a specific category. This is achieved by making use of the global average pooling strategy of [Lin et al., 2013], and importantly, without using any bounding box, or pixel-level annotations.

In our case, as discussed in section 3.4.2.2, our starting point is a fully-convolutional version of the VGG-16 network. Just before the final output layer (the cross entropy loss layer for multi class categorization), we perform global average pooling on the convolutional feature maps and use the resulting features as input to a fully-connected layer that produces class scores. Specifically, let $f_k(x, y)$ denote the activation of unit k at spatial location (x, y) in the last convolutional layer, and $F^k = \sum_{x,y} f_k(x, y)$ the result of global average pooling for unit k. Then, the predicted score for a given class c can be written as $S_c = \sum_k w_k^c F^k$, where w_k^c is the weight corresponding to class c for unit k. In essence, w_k^c indicates the importance of unit k for class c.

To generate a CAM, one can thus rely on these weights. In particular, these weights are used in a linear combination of the activations of the units in the last convolutional layer. This lets us express a CAM for class c as

$$M_c(x, y) = \sum_k w_k^c f_k(x, y) \ . \tag{3.1}$$

Ultimately, $M_c(x, y)$ directly indicates how important the observations at spatial grid (x, y) are to classify the input image as belonging to class c.

As can be seen in Figure 3.5, the resulting CAMs suffer from two main drawbacks. First, they only roughly match the shape of the object, yielding inaccurate localization of the object's boundary. Second, they typically only focus on the discriminative parts of the objects, which is sufficient for object recognition, but not for segmentation. To overcome these limitations, we propose to combine these CAMs with our foreground/background masks, to obtain more accurate and more complete multi-class masks.

To this end, and as suggested in [Zhou et al., 2016], we first generate binary masks from each M_c by setting to 1 the values that are above 20% of the maximum value in each M_c, and to 0 the other ones. Let us denote by B_c the resulting binary mask for class c. From these binary masks and the foreground/background probabilities P_f obtained by fusing the activations of the fourth and fifth convolutional layers, we form a new multi-class mask, which, for each class c, is defined as a map

$$Q_c = P_f \odot B_c \ , \tag{3.2}$$

where we think of each map as a matrix, and where $\odot$ indicates the Hadamard (elementwise) product. This, in essence, can be thought of as a class-specific truncated version of P_f, where the truncation masks are obtained from the M_cs, with a

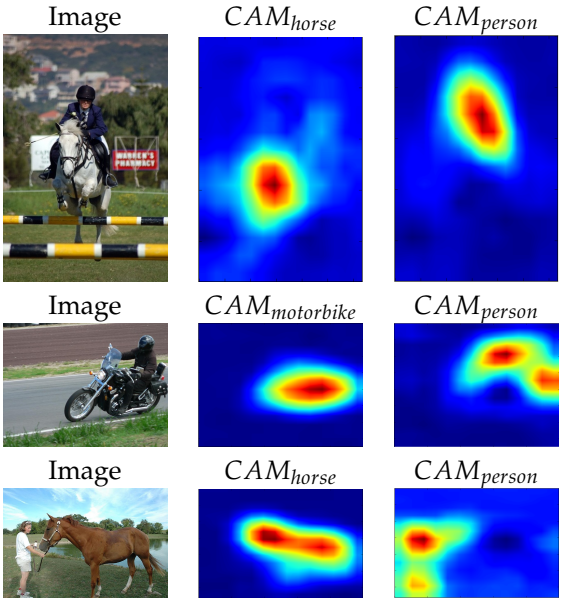


Figure 3.5: CAM for each class obtained by the localization network.

permissive threshold of 20% to avoid cutting out too many regions.

To obtain our final multi-class masks, we combine these class-specific truncated fusion maps with the original CAMs. To this end, we make use of a linear combination, which yields, for each class c, the final map

$$P_c = \alpha \cdot Q_c + (1 - \alpha) \cdot M_c \ , \tag{3.3}$$

where, in practice, we set $\alpha = 0.5$, and which is normalized to obtain a probability. The resulting probabilities are compared to the fusion-based ones and to the CAMs in Figure 3.6. Note that the final maps preserve the more accurate boundary information and the better object coverage of the fusion-based ones, while removing their noise, thanks to the CAMs.

At this point, we have probability maps for each foreground class c, but not for the background class. To generate such a background map, we simply use the probabilities of the locations that have not been considered as foreground classes in M_c. To this end, we define

$$M_0 = 1 - \frac{1}{C} \sum_c M_c \ , \tag{3.4}$$

which, in turn, lets us define the background map as

$$P_0 = \alpha \cdot (1 - P_f) + (1 - \alpha) \cdot M_0. \tag{3.5}$$

While better than both our foreground/background masks and the CAMs, our multi-class masks remain noisy. To address this, in the next section, we propose to make use of a fully-connected CRF with higher-order terms.

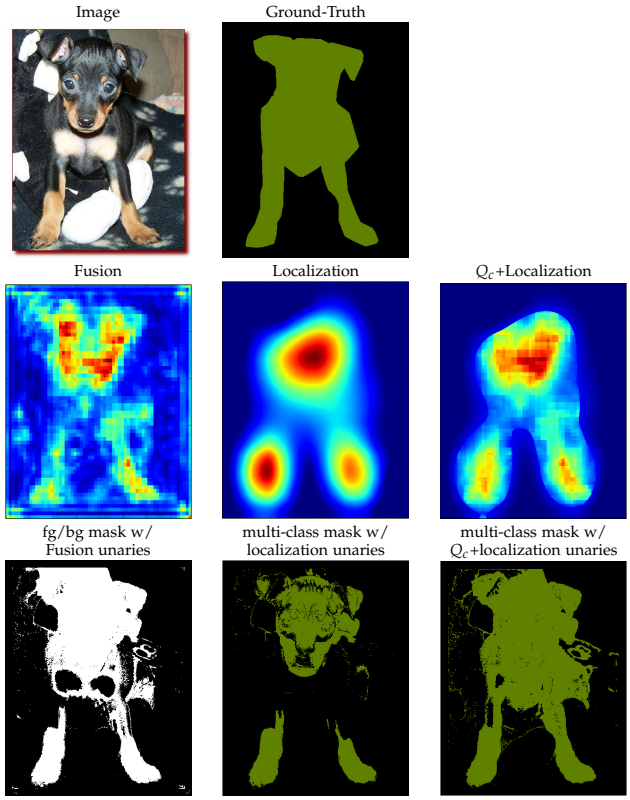


Figure 3.6: Effect of adding localization information to our Fusion map (Q_c).

3.3.1.3 Smoothing the Masks with a Dense CRF

To smooth out initial noisy masks, we make use of a fully-connected CRF with higher-order terms. Note that, while we consider the general, multi-class case, the formalism discussed below applies to both our foreground/background masks and to our multi-class masks.

Specifically, let $\mathbf{x} = \{x_i\}_{i=1}^{W \cdot H}$ be the set of random variables, where x_i encodes the label of pixel i, i.e., either one of the foreground classes or background. We encode the joint distribution over all pixels with a Gibbs energy of the form

$$E(\mathbf{x} = \mathbf{X}) = -\sum_i \theta_i(x_i = X_i) + \sum_i \sum_{j>i} \theta_{ij}(x_i = X_i, x_j = X_j) + \sum_{\mathbf{x}_s \in \mathcal{R}} \theta_s(\mathbf{x}_s = \mathbf{X}_s)\ , \quad (3.6)$$

where θ_i is a unary potential defining the cost of assigning label X_i to pixel i, and the second and third terms encode pairwise and higher-order potentials, respectively, with $\mathcal{R}$ a set of regions.

The unary potential is obtained directly from the probability maps introduced in

either Section 3.3.1.1 or 3.3.1.2 as

$$\theta_i(x_i = X_i) = -\log\left(\frac{\exp\left(P(x_i = X_i)\right)}{\sum_{l=1}^{C}\exp\left(P(x_i = l)\right)}\right), \tag{3.7}$$

where P can be either P_f or P_c.

The pairwise potential θ_{ij} encodes the compatibility of a joint label assignment for two pixels. Following [Krähenbühl and Koltun, 2011], we define this pairwise term as a contrast-sensitive Potts model using two Gaussian kernels encoding color similarity and spatial smoothness. Such a model penalizes two pixels at relatively close spatial locations and with similar appearance to be assigned different labels.

For the higher-order terms, we make use of a P^n-Potts model encouraging all the pixels in one region to be assigned the same label. To define the regions, we propose to make use of the crisp boundary detection algorithm of [Isola et al., 2014]. This algorithm aims at detecting the boundaries between semantically different objects visible in the scene. It is based on a simple underlying principle: pixels belonging to the same object exhibit higher statistical dependencies than pixels belonging to different objects. This method is unsupervised and can adapt to each input image independently. As illustrated in Figure 3.7, the resulting crisp boundaries can be thought of as defining semantically coherent regions, which are thus very well-suited to our goal. For each region $\mathbf{x}_s$, we then define the cost of the higher-order term as

$$\theta_s(\mathbf{x}_s = l) = -\log\left(\frac{\sum_{x_i \in \mathbf{x}_s} P(x_i = l)}{N_s}\right), \tag{3.8}$$

if all the pixels are assigned the same label l, and a maximum cost otherwise. Here, N_s indicates the number of pixels in region s.

By using Gaussian pairwise potentials and P^n-Potts higher-order ones, we can make use of the inference strategy of [Vineet et al., 2014], which relies on the filtering-based mean-field method of [Krähenbühl and Koltun, 2011]. In Figs. 3.3–3.7, we show the effect of CRF smoothing on our masks with and without higher-order terms.

3.3.2 Weakly-Supervised Learning

We now introduce our learning algorithm for weakly-supervised semantic segmentation. We first introduce a simple loss based on image tags only, and then show how we can incorporate our two different types of masks in this framework.

Intuitively, given image tags, one would like to encourage the image pixels to be labeled as one of the classes that are observed in the image, while preventing them to be assigned to unobserved classes. Note that this assumes that the tags cover all the classes depicted in the image. This assumption, however, is commonly employed in weakly-supervised semantic segmentation [Bearman et al., 2016; Pathak et al., 2015b; Pinheiro and Collobert, 2015]. Formally, given an input image I, let $\mathcal{L}$ be the set of classes that are present in the image (including background) and $\bar{\mathcal{L}}$ the set of classes that are absent. Furthermore, let us denote by $s_{i,j}^k(\theta)$ the score produced by our

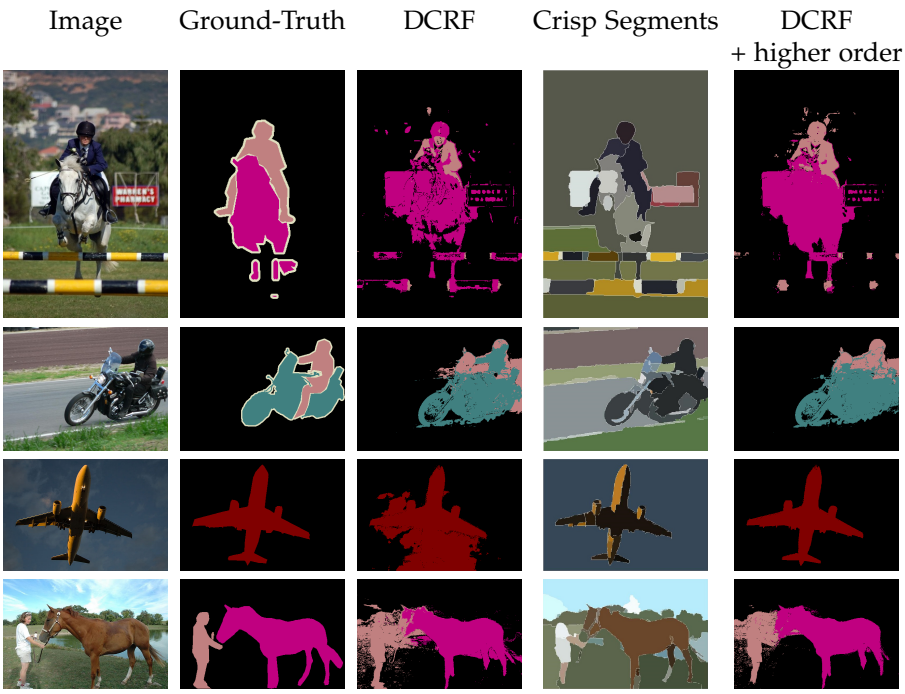


Figure 3.7: Effect of using higher-order potentials using regions obtained by the crisp boundary detection method of [Isola et al., 2014].

network with parameters θ for the pixel at location (i, j) and for class k, $0 \leq k < N$. Note that, in general, we will omit the explicit dependency of the variables on the network parameters. Finally, let $S^k_{i,j}$ be the probability of class k obtained after a softmax layer, i.e.,

$$S^k_{i,j} = \frac{\exp(s^k_{i,j})}{\sum_{c=1}^{N} \exp(s^c_{i,j})} \,. \tag{3.9}$$

Encoding the above-mentioned intuition can then simply be achieved by designing a loss of the form

$$L_{weak} = -\frac{1}{|\mathcal{L}|} \sum_{k \in \mathcal{L}} \log S^k - \frac{1}{|\bar{\mathcal{L}}|} \sum_{k \in \bar{\mathcal{L}}} \log(1 - S^k) \,, \tag{3.10}$$

where S^k represents a candidate score for each class in the image. In short, the first term in Equation 3.10 expresses the fact that the present classes should be in the image, while the second term penalizes the pixels that have high probabilities for the absent classes. In practice, instead of computing S^k as the maximum probability (as previously used in [Pathak et al., 2015b; Bearman et al., 2016]) for class k over all pixels in the image, we make use of the convex Log-Sum-Exp (LSE) approximation of the maximum (as previously used in [Pinheiro and Collobert, 2015]), which can

be written as

$$S^k = \frac{1}{r} \log \left[\frac{1}{|I|} \sum_{i,j \in I} \exp(r S^k_{i,j}) \right] , \tag{3.11}$$

where $|I|$ denotes the total number of pixels in the image and r is a parameter allowing this function to behave in a range between the maximum and the average. In practice, following [Pinheiro and Collobert, 2015], we set r to 5.

The loss in Equation 3.10 does not rely on any notion of foreground and background. As a consequence, minimizing it will typically yield poor object localization accuracy. To overcome this issue, we propose to make use of our built-in priors introduced in Sections 3.3.1.1 and 3.3.1.2. Below, we start with the foreground/background case, and then turn to the multi-class scenario.

3.3.2.1 Incorporating Foreground/Background Masks

When only a foreground/background probability is available, we cannot directly reason at the level of specific classes. Instead, we rely on this mask to encourage all pixels labeled as one of the object tags to lie within a foreground region, while the other pixels should belong to the background.

To this end, let $M_{i,j}$ denote the mask value at pixel (i, j), i.e., $M_{i,j} = 1$ if pixel (i, j) belongs to the foreground and 0 otherwise. We can then re-write our loss as

$$L_{mask} = -\frac{1}{|\mathcal{L}| - 1} \sum_{k \in \mathcal{L}, k \neq 0} \log(S^k) - \log(S^0) - \frac{1}{|\bar{\mathcal{L}}| \cdot |I|} \sum_{i,j \in I,\, k \in \bar{\mathcal{L}}} \log(1 - S^k_{i,j}) ,$$

where

$$S^k = \frac{1}{r} \log \left[\frac{1}{|M|} \sum_{i,j | M_{i,j}=1} \exp(r S^k_{i,j}) \right] , \tag{3.12}$$

and

$$S^0 = \frac{1}{r} \log \left[\frac{1}{|\bar{M}|} \sum_{i,j | M_{i,j}=0} \exp(r S^0_{i,j}) \right] , \tag{3.13}$$

with $|M|$ and $|\bar{M}|$ the number of foreground and background pixels, respectively. S^k computes an approximate maximum probability for the present class k over all pixels in the foreground mask. Similarly, S^0 denotes an approximate maximum probability for the background class over all pixels outside the foreground mask. In short, the loss of Equation 3.12 favors present classes to appear in the foreground mask, while pixels predicted as background should be assigned to the background class and no pixels should take on an absent label.

To learn the parameters of our network, we follow a standard back-propagation strategy to search for the parameters θ that minimize the loss in Equation 3.12. In particular, the network is fine-tuned using stochastic gradient descent (SGD) with momentum to update the weights by a linear combination of the negative gradient and the previous weight update. At inference time, given the test image, the network

performs a dense prediction. We optionally apply a fully-connected CRF with higher-order terms similar to the one discussed above to smooth the segmentation. To this end, we used the default CRF parameter values as in the original paper [Krähenbühl and Koltun, 2011].

3.3.2.2 Incorporating Multi-class Masks

In the presence of multi-class masks, we can then reason about the specific classes that are observed in the input image. In this scenario, we would like to encourage the pixels set to 1 in one particular class mask corresponding to one input tag to be assigned the label of this class. Enforcing this strongly, e.g., by considering the maximum score over all pixels in a mask, would unfortunately be sensitive to noise in the mask, as further discussed in our experiments. Instead, here, we propose to again make use of the LSE to have a softer penalty.

Specifically, let M_k be the mask corresponding to the image tag, i.e., class label, k. We propose to take into account our multi-class masks by re-writing our loss function as

$$L_{weak} = -\frac{1}{|\mathcal{L}|}\sum_{k\in\mathcal{L}}\log(S^k) - \frac{1}{|\bar{\mathcal{L}}|\cdot|I|}\sum_{i,j\in I,\, k\in\bar{\mathcal{L}}}\log(1-S^k_{i,j})\;, \tag{3.14}$$

where

$$S^k = \frac{1}{r}\log\left[\frac{1}{|M_k|}\sum_{i,j|M_k=1}\exp(rS^k_{i,j})\right]\;. \tag{3.15}$$

In other words, this loss encourages, for each present class k, including the background class, the pixels belonging to the corresponding mask to be assigned label k, while penalizing the pixels that take on an absent label. We use the same learning strategy as in the foreground/background case to minimize this. Furthermore, as before, during inference, the network provides a dense labeling for an input test image, without requiring any tag, and this labeling can optionally be smoothed via CRF inference.

3.4 Experiments

In this section, we first describe the datasets used for our experiments, and then provide details about our learning and inference procedures. We then compare our method with foreground/background masks and with multi-class ones to the state-of-the-art weakly supervised semantic segmentation algorithms. Finally, we provide a thorough evaluation of the effect of the different components of our approach.

3.4.1 Datasets

PASCAL VOC 2012. In our experiments, we made use of the standard Pascal VOC 2012 dataset [Everingham et al., 2015], which serves as a benchmark in most weakly-supervised semantic segmentation papers [Bearman et al., 2016; Papandreou et al.,

2015; Pathak et al., 2015b; Pinheiro and Collobert, 2015; Pathak et al., 2015a]. This dataset contains 21 classes, and 10,582 training images (the VOC 2012 training set and the additional data annotated by [Hariharan et al., 2011]), 1,449 validation images and 1,456 test images. The image tags were obtained from the pixel-level annotations by simply listing the classes observed in each image. As in [Bearman et al., 2016; Papandreou et al., 2015; Pinheiro and Collobert, 2015; Pathak et al., 2015a], we report results on both the validation and the test set.

YouTube Objects. This dataset (YTO) [Prest et al., 2012] contains videos collected from YouTube by querying for the names of 10 object classes of the PASCAL VOC dataset. It contains between 9 and 24 videos per class. For our experiments, we uniformly extracted around 2200 frames per class to obtain a total of 22k frames out of 700k available in the dataset. For evaluation we use the subset of images with pixel-level annotations provided by [Jain and Grauman, 2014b]. Note that there is no overlap between this subset and the shots from which we extracted the training data.

Microsoft COCO. For MS-COCO [Lin et al., 2014], we made use of 80k training samples with only image tags to train our network and 40k validation samples to evaluate the performance of our method. The MS-COCO annotations were designed for instance level labeling. As such, some pixels in the images can be assigned multiple labels. For example, a pixel can belong to both Fork and Dining Table. To evaluate our results for semantic segmentation, we obtained a unique ground-truth label per pixel by using the label of the smallest object, that is, fork in the example above.

Note that Sections 3.4.3.1 and 3.4.3.2 focus on the PASCAL VOC dataset, which is the one commonly used for weakly-supervised semantic segmentation. The results for YTO and MS-COCO, which demonstrate the generality of our method, are provided in Section 3.4.3.3.

3.4.2 Implementation Details

3.4.2.1 Semantic Segmentation Networks

As most recent weakly-supervised semantic segmentation algorithms [Bearman et al., 2016; Papandreou et al., 2015; Pathak et al., 2015b; Pinheiro and Collobert, 2015; Pathak et al., 2015a; Kolesnikov and Lampert, 2016], our architecture is based on the VGG-16 network [Simonyan and Zisserman, 2014], whose weights were trained on ImageNet for the task of object recognition. Following the fully-convolutional approach [Long et al., 2015], all fully-connected layers are converted to convolutional layers, and the final classifier replaced with a 1×1 convolution layer with N channels, where N represents the number of classes of the problem. We use two different versions of this fully-convolutional network. When utilizing foreground/background masks, inspired by [Chen et al., 2014], we used a stride of 8 and a relatively small receptive field of 128 pixels, which has proven effective in practice for weakly-supervised semantic segmentation [Papandreou et al., 2015]. By contrast, when using multi-class masks, inspired by [Chen et al., 2014] again, we found that using a larger field of view improves the results. We therefore employed a kernel size of 3×3 in

the convolutional layer corresponding to the first fully connected layer of VGG-16 and an input stride of 12, resulting in a receptive field size of 224. We also reduced the number of filters from 4096 to 1024 to allow for faster training [Chen et al., 2014]. With both types of masks, at the end of the network, we added a deconvolution layer to up-sample the output of the network to the size of the input image. In short, the network takes an image of size $W \times H$ as input and generates an $N \times W \times H$ output encoding a score for each pixel and for each class.

For both types of masks, the parameters of the network were found using stochastic gradient descent with a learning rate of 10^{-4} for the first 40k iterations and 10^{-5} for the next 20k iterations, a momentum of 0.9, a weight decay of 0.0005, and mini-batches of size 1. Similarly to recent weakly-supervised segmentation methods [Pinheiro and Collobert, 2015; Pathak et al., 2015a,b; Bearman et al., 2016; Papandreou et al., 2015], the network weights were initialized with those of a network pre-trained for a 1000-way classification task on the ILSVRC 2012 dataset [Russakovsky et al., 2015]. Hence, for the last convolutional layer, we used the weights corresponding to the 20 classes shared by Pascal VOC and ILSVRC. For the background class, we initialized the weights with zero-mean Gaussian noise with a standard deviation of 0.1.

At inference time, given the test image, but no tags, the network generates a dense prediction as a complete semantic segmentation map. We used C++ and Python (Caffe framework [Jia et al., 2014]) for our implementation. As other methods [Pathak et al., 2015a; Papandreou et al., 2015; Kolesnikov and Lampert, 2016], we further optionally apply a dense CRF to refine this initial segmentation. As mentioned in Section 3.3.1.3, we added higher-order potentials to the dense pairwise CRF.

3.4.2.2 Localization Network

For the localization network, we followed the approach introduced in [Zhou et al., 2016]. Specifically, the architecture of the network was again derived from the VGG-16 architecture [Simonyan and Zisserman, 2014], pre-trained for the task of object recognition on ImageNet. We then substituted the last two fully-connected layers, fc6 and fc7, with randomly initialized convolutional layers. The output of the last convolutional layer acts as input to a global average pooling layer followed by a fully-connected prediction layer corresponding to the number of foreground classes of interest (20 for PASCAL VOC). The network was fine-tuned for object recognition on the training set of the PASCAL VOC 2012 dataset with a cross entropy loss. To this end, we used images of size of 224×224 as input, and mini-batches of size 15. The other optimization parameters were set to the same values as for the semantic segmentation network.

Note that we could in principle also fine-tune the VGG-16 network used to generate our foreground/background masks for object recognition on the target dataset (e.g., PASCAL VOC). In practice, however, we observed that this did not improve the quality of our masks.

Table 3.1: Per class IOU on the PASCAL VOC 2012 validation set for methods trained using image tags.

Method	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIOU
MIL(Tag) [Pinheiro and Collobert, 2015]	37.0	10.4	12.4	10.8	5.3	5.7	25.2	21.1	25.15	4.8	21.5	8.6	29.1	25.1	23.6	25.5	12.0	28.4	8.9	22.0	11.6	17.8
MIL(Tag) w/ILP [Pinheiro and Collobert, 2015]	73.2	25.4	18.2	22.7	21.5	28.6	39.5	44.7	46.6	11.9	40.4	11.8	45.6	40.1	35.5	35.2	20.8	41.7	17.0	34.7	30.4	32.6
MIL(Tag) w/ILP+sspxl [Pinheiro and Collobert, 2015]	77.2	37.3	18.4	25.4	28.2	31.9	41.6	48.1	50.7	12.7	45.7	14.6	50.9	44.1	39.2	37.9	28.3	44.0	19.6	37.6	35.0	36.6
What's the point(Tag) W/Obj [Bearman et al., 2016]	78.8	41.6	19.8	38.7	33.0	17.2	33.8	38.8	45.0	10.4	35.2	12.6	42.3	34.3	33.2	22.7	18.6	40.1	14.9	37.7	28.1	32.2
EM-Fixed(Tag)+CRF [Papandreou et al., 2015]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	20.8
EM-Adapt(Tag)+CRF [Papandreou et al., 2015]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	38.2
CCNN(Tag) [Pathak et al., 2015a]	66.3	24.6	17.2	24.3	19.5	34.4	45.6	44.3	44.7	14.4	33.8	21.4	40.8	31.6	42.8	39.1	28.8	33.2	21.5	37.4	34.4	33.3
CCNN(Tag)+CRF [Pathak et al., 2015a]	68.5	25.5	18.0	25.4	20.2	36.3	46.8	47.1	48.0	15.8	37.9	21.0	44.5	34.5	46.2	40.7	30.4	36.3	22.2	38.8	36.9	35.3
SEC+CRF [Kolesnikov and Lampert, 2016]	82.2	61.7	26.0	60.4	25.6	45.6	70.9	63.2	72.2	20.9	52.9	30.6	62.8	56.8	63.5	57.1	32.2	60.6	32.3	44.8	42.3	50.7
DCSM+CRF [Shimoda and Yanai, 2016]	76.7	45.1	24.6	40.8	23.0	34.8	61.0	51.9	52.4	15.5	45.9	32.7	54.9	48.6	57.4	51.8	38.2	55.4	32.2	42.6	39.6	44.1
fg/bg masks+CRF	79.2	60.1	20.4	50.7	41.2	46.3	62.6	49.2	62.3	13.3	49.7	38.1	58.4	49.0	57.0	48.2	27.8	55.1	29.6	54.6	26.6	46.6
Ours fg/bg masks+CRF+H.O	82.0	68.0	26.9	66.5	34.1	47.4	57.3	51.7	72.2	14.5	50.6	26.6	65.3	55.9	58.7	25.8	29.7	62.5	27.9	54.1	30.0	48.0
Ours multi-class masks+CRF	82.2	59.5	27.4	66.7	25.2	44.1	71.1	55.1	71.9	19.7	52.3	36.7	65.6	59.4	62.8	55.3	32.3	65.5	34.3	43.4	38.8	50.9

Table 3.2: Per class IOU on the PASCAL VOC 2012 test set for methods trained using image tags.

Method	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIOU
CCNN (tags)+CRF [Pathak et al., 2015a]	-	24.2	19.9	26.3	18.6	38.1	51.7	42.9	48.2	15.6	37.2	18.3	43.0	38.2	52.2	40.0	33.8 -	36.0	21.6	33.4	38.3	35.6
MIL-FCN [Pinheiro and Collobert, 2015]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	25.7
MIL-sppxl [Pinheiro and Collobert, 2015]	74.7	38.8	19.8	27.5	21.7	32.8	40.0	50.1	47.1	7.2	44.8	15.8	49.4	47.3	36.6	36.4	24.3	44.5	21.0	31.5	41.3	35.8
MIL-obj [Pinheiro and Collobert, 2015]	76.2	42.8	20.9	29.6	25.9	38.5	40.6	51.7	49.0	9.1	43.5	16.2	50.1	46.0	35.8	38.0	22.1	44.5	22.4	30.8	43.0	37.0
EM-Adapt+CRF [Papandreou et al., 2015]	76.3	37.1	21.9	41.6	26.1	38.5	50.8	44.9	48.9	16.7	40.8	29.4	47.1	45.8	54.8	28.2	30.0	44.0	29.2	34.3	46.0	39.6
SEC+CRF [Kolesnikov and Lampert, 2016]	83.0	55.6	27.4	61.1	22.9	52.4	70.2	58.8	70.0	22.1	54.3	27.9	67.4	59.4	70.7	59.0	38.7	58.6	38.1	37.6	45.2	51.5
DCSM+CRF [Shimoda and Yanai, 2016]	78.1	43.8	26.3	49.8	19.5	40.3	61.6	53.9	52.7	13.7	47.3	34.8	50.3	48.9	69.0	49.7	38.4	57.1	34.0	38.0	40.0	45.1
fg/bg masks+CRF	80.3	57.5	24.1	66.9	31.7	43.0	67.5	48.6	56.7	12.6	50.9	42.6	59.4	52.9	65.0	44.8	41.3	51.1	33.7	44.4	33.2	48.0
Ours fg/bg masks+CRF(H.O)	83.4	65.4	29.0	68.5	33.4	51.6	58.4	53.5	68.3	15.7	54.1	30.2	66.9	57.9	66.0	23.7	39.6	61.6	29.7	51.9	31.8	49.6
Ours multi-class masks+CRF(H.O)	83.5	60.8	29.8	66.6	23.2	52.1	69.3	53.8	70.4	19.1	56.8	40.1	71.0	59.7	71.4	54.9	33.9	71.2	40.5	35.4	41.9	52.6

Table 3.3: Mean IOU on the PASCAL VOC validation and test sets for other methods trained with higher level of supervision or additional training data. Note that, while our approach requires no additional supervision or training data, its accuracy is comparable to or higher than that of other methods.

Methods	mIoU(val)	mIOU(test)
[Pinheiro and Collobert, 2015]: MIL(Tag) w/ILP+bbox	37.8	37.0
[Pinheiro and Collobert, 2015]: MIL(Tag) w/ILP+seg	42.0	40.6
[Wei et al., 2016a]: SN-B+MCG seg	41.9	43.2
[Bearman et al., 2016]: 1Point	35.1	-
[Bearman et al., 2016]: Objectness+1Point	42.7	-
[Bearman et al., 2016]: Objectness+1Point(GT)	46.1	-
[Bearman et al., 2016]: Objectness+AllPoints (weighted)	43.4	-
[Bearman et al., 2016]: Objectness+1 squiggle per class	49.1	-
[Pathak et al., 2015a]: Random Crops+CRF	36.4	-
[Pathak et al., 2015a]: Size Info.+CRF	42.4	45.1
[Wei et al., 2016b]: STC + CRF + additional train data	49.8	51.2
[Wei et al., 2016a] SN-B+MCG seg	41.9	43.2
[Qi et al., 2016]: Augmented feedback+MCG+CRF	54.3	55.5
CheckMask procedure+CRF	51.5	52.9
Ours (fg/bg masks)+CRF(H.O)	48.0	49.6
Ours (multi-class masks)+CRF(H.O)	50.9	52.6

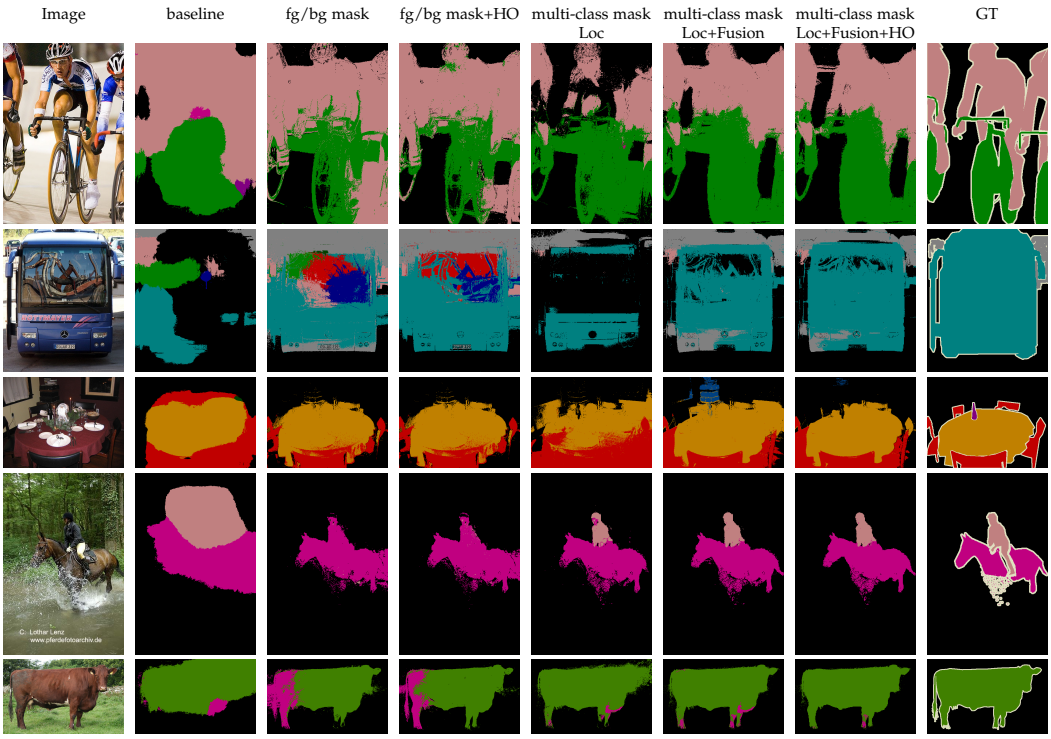


Figure 3.8: Qualitative results from the Pascal VOC validation set.

3.4.3 Experimental Results

Below, we first compare our approach with state-of-the-art baselines on PASCAL VOC. Then, we evaluate the different components of our method on the validation set of the PASCAL VOC dataset. Finally, we provide results of our complete framework on two more datasets to show the generality of our approach.

3.4.3.1 Comparison to State-of-the-art

We first compare our approach with state-of-the-art baselines on PASCAL VOC. To this end, we report the Intersection over Union (IOU), which is the most commonly used metric for semantic segmentation. In the following, we refer to our approach with foreground/background masks as *Ours fg/bg masks* and with multi-class masks as *Ours multi-class masks*.

We report the results of our approach and the state-of-the-art methods relying on tags only in Table 3.1 and Table 3.2 for the Pascal VOC 2012 validation and test images, respectively. Note that our approach, with either type of masks, outperforms most of the baselines by a large margin. The only exception is the contemporary SEC algorithm of [Kolesnikov and Lampert, 2016], which outperforms the foreground/background version of our method. Note that SEC also relies on the multi-class results of the localization network. We believe that the fact that our multi-class-based approach performs slightly better than SEC, particularly on the test images, indicates the effectiveness of our combination of the localization network with our fusion-based built-in prior. Importantly, the results also show that we outperform the methods based on an objectness prior [Bearman et al., 2016; Pinheiro and Collobert, 2015], which evidences the benefits of using our built-in foreground/background masks instead of external objectness algorithms. Note that our results with foreground/background masks has been reported with and without higher-order potentials.

We then compare our approach, which uses only image tags, with other methods that rely on additional training data or additional supervision. In particular, these include the point supervision of [Bearman et al., 2016], the random crops of [Papandreou et al., 2015], the size information of [Pathak et al., 2015a], the MCG segments of [Pinheiro and Collobert, 2015; Wei et al., 2016a; Qi et al., 2016], additional training data of [Wei et al., 2016b], and the proposed CheckMask procedure on foreground/background masks. The results of this comparison are provided in Table 3.3. Note that, with the exception of our own CheckMask procedure and the method of [Qi et al., 2016], which uses MCG segments, our approach with multi-class masks outperforms all the baselines, and with foreground/background masks most of them, despite the fact that we do not require any supervision other than tags. It is worth mentioning that other approaches have proposed to rely on labeled bounding boxes, which require a user to provide a bounding box for each individual foreground object in an image and to associate a label to each such bounding box. While this procedure is clearly costly, we achieve accuracies close to these baselines (52.5% for [Papandreou et al., 2015] when using labeled bounding boxes and 54.1%

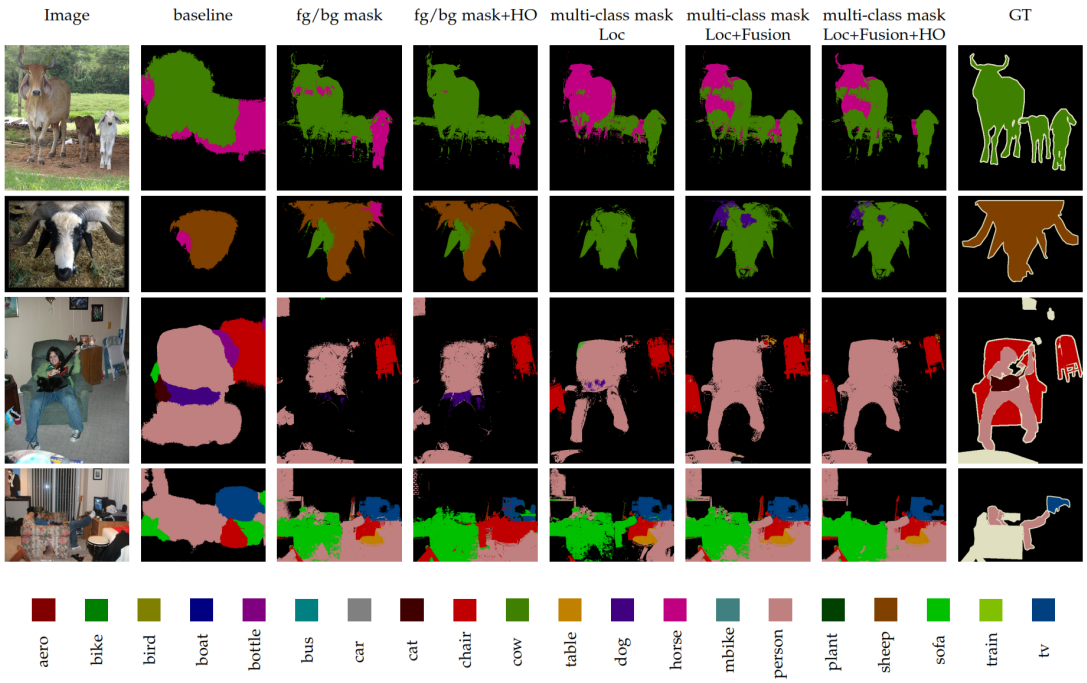


Figure 3.9: Failure cases from the Pascal VOC validation set.

for [Papandreou et al., 2015] when using labeled bounding boxes in an EM process vs 50.9% for our approach with image tags only). We believe that this further evidences the benefits of our approach.

In Figs. 3.8 and 3.9, we show some successful segmentations and failure cases of our approach, respectively. In some cases (e.g., first row of Figure 3.9), these failures are due to the output scores of the network, which are used in Eqs. 3.12 and 3.14. Other failures are due to errors in our predicted masks. For example, the second row of Figure 3.9 indicates that errors appear after using the localization network to generate multi-class masks. The most common type of failure occurs in the presence of complex scenes, in which the network is unable to segment small objects. The last two rows of Figure 3.9 show some of these cases.

3.4.3.2 Ablation Study

We now study the effect of the different components of our approach on our results. In particular, we first evaluate our predicted masks, and then discuss semantic segmentation results.

Mask Evaluation: Foreground/background To evaluate our foreground/background masks, we made use of 10% of randomly chosen training images from the Pascal VOC dataset. We then generated foreground/background masks for these images using our approach, which relies on the activations of the fourth and fifth layers

Table 3.4: Comparison of our foreground/background masks with those obtained using the objectness methods of [Alexe et al., 2012] and [Arbeláez et al., 2014].

	Mean IoU
Masks obtained using [Alexe et al., 2012]	52.34%
Masks obtained using [Arbeláez et al., 2014]	50.20%
Our masks	**60.08**%

of the segmentation network pre-trained on ImageNet (i.e., before fine-tuning it for semantic segmentation). These masks can then be compared to ground-truth foreground/background masks obtained directly from the pixel level annotations.

We compare our masks with the objectness criteria of [Alexe et al., 2012] and [Arbeláez et al., 2014], which were employed for the purpose of weakly-supervised semantic segmentation by [Bearman et al., 2016] and [Pinheiro and Collobert, 2015; Wei et al., 2016a; Qi et al., 2016], respectively.

Note that some objectness methods, such as [Cheng et al., 2014; Arbeláez et al., 2014], that have been used for weakly-supervised semantic segmentation [Pinheiro and Collobert, 2015; Dai et al., 2015; Wei et al., 2016a; Qi et al., 2016], require training data with pixel-level or bounding box annotations, and thus are not really comparable to our approach. Note also that a complete evaluation of objectness methods goes beyond the scope of this chapter, which focuses on weakly-supervised semantic segmentation.

The objectness methods of [Alexe et al., 2012] and [Arbeláez et al., 2014] produce a per-pixel foreground probability map. For our comparison to be fair, we further refined these maps using the same dense CRF as in our approach. In Table 3.4, we provide the results of these experiments in terms of mean Intersection Over Union (mIOU) with respect to the ground-truth masks. Note that our masks are more accurate than those of [Alexe et al., 2012; Arbeláez et al., 2014]. In particular, we have found that our masks yield a much better object localization accuracy. In Figure 3.10, we show some qualitative results of these three approaches. Note that this further evidences the benefits of our foreground/background masks. In particular, our masks yield a much better object localization accuracy.

Mask Evaluation: Multi-class As discussed before, our multi-class masks rely, in part, on the localization network. Although the localization map provides useful information about the location of the objects, it is not sufficient on its own to generate accurate masks. In addition to its lack of accuracy at the object's boundary and the incompleteness of its segmentation, as illustrated earlier in Figure 3.5, the accuracy of the localization network varies greatly for different classes. We illustrate this in Figure 3.11 for the successful case of the monitor class and for the failure case of potted plants. In the case of monitor, which, most of the time, is located in the center of the image (see Average on GT in 3.11), the network is able to localize it reasonably well. By contrast, potted plants are scattered in all locations in the dataset (see Average on GT), and the network therefore fails to localize it accordingly. As a matter

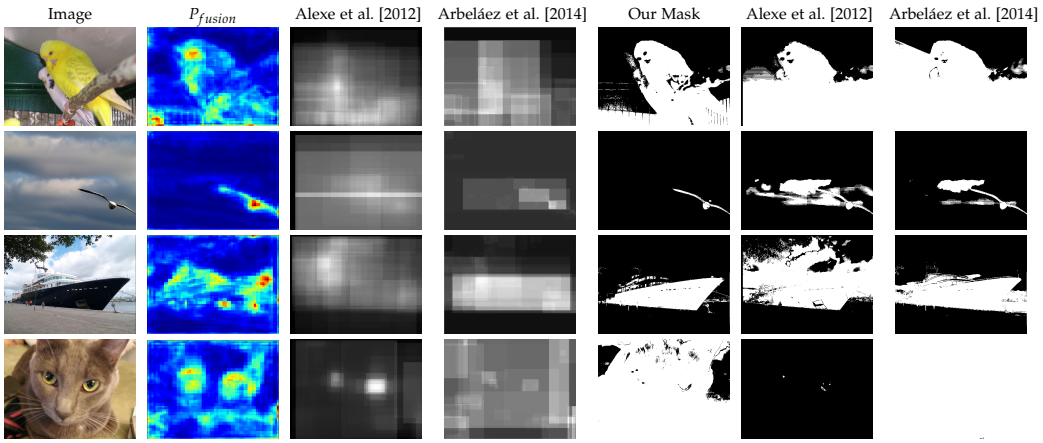


Figure 3.10: Qualitative comparison of our masks with those of Objectness Map of Alexe et al. [2012] and MCG Map of Arbeláez et al. [2014]. Note that our approach yields much better localization accuracy.

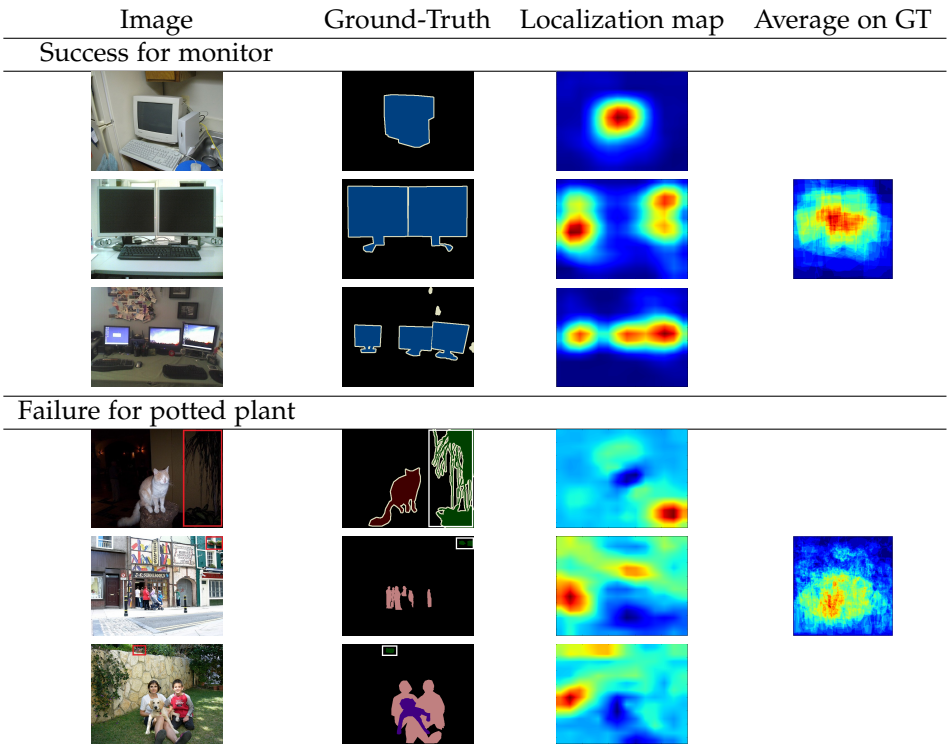


Figure 3.11: Success and failure cases of the localization network.

of fact, when training our method with masks obtained from the localization network only, the IOU of potted plants is 0. This IOU increases to 32.3 when combining the localization network with our fusion-based masks, as discussed in Section 3.3.1.2.

Since our method generates multi-class masks, one could think of directly using these masks to obtain the final semantic segmentation of an input image, that is,

Table 3.5: Accuracy of the multi-class masks when directly used for segmentation (without any network), assuming known tags at test time.

Methods	Mean IoU
multi-class masks using localization	43.0
multi-class masks using localization+fusion	46.6

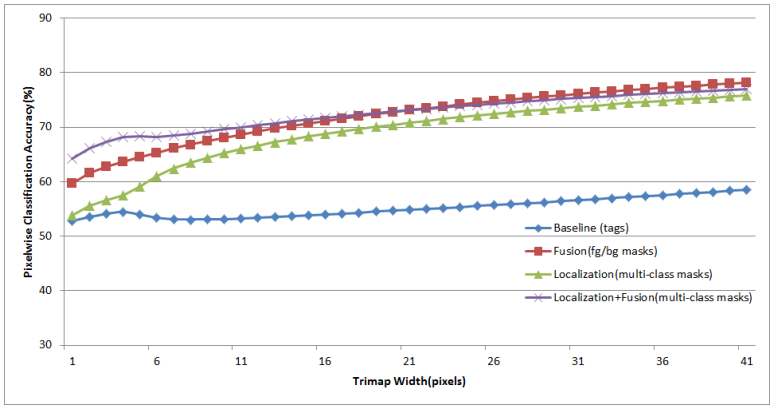


Figure 3.12: Pixel classification accuracy as a function of the bandwidth around the object boundaries on the Pascal VOC validation set. Note that using our fusion-based masks helps improving the accuracy at the boundary of the objects.

without training a network at all. We evaluated how well this naive approach performs on the Pascal VOC validation data. To further help this baseline, we made use of the ground-truth tags to filter out the absent classes from the masks' predictions. The results of this experiment are reported in Table 3.5 for the localization masks only and for our multi-class masks. Note that these results, despite relying on ground-truth tags at test time, are lower than that of our approach, which does not use this information. This confirms the importance of training a network based on our masks, rather than directly using the masks for prediction.

To evaluate the accuracy of the different types of masks at the boundary of the objects, we further made use of the Trimap accuracy [Kohli et al., 2009], which focuses on the segmentation error within a region around the true boundaries. Specifically, we evaluate the quality of a segmentation by counting the number of pixels misclassified in the region surrounding the actual object boundary and not over the entire image. The error is computed for different widths of the evaluation region. In Figure 3.12, we report the Trimap accuracy as a function of the width of the region around the boundary for the results obtained with our fusion-based foreground/background masks, the localization network masks, and our multi-class masks (fusion+localization).

In addition to this, we also report the error of a simple baseline consisting of not using any mask, but only the tags, i.e., using Equation 3.10 as training loss. Note that using masks clearly improves boundary accuracy, particularly when using our fusion-based masks, with or without the additional localization ones. Recall, how-

Table 3.6: Mean IOU on PASCAL VOC val. set for different setups of our method.

Methods	mIOU
Tag-only Baseline (no mask)	31.0
Foreground/Background Priors	47.3
Foreground/Background Priors + CheckMask procedure	51.5
Foreground/Background Priors + Higher Order	48.0
Localization Priors	45.9
Localization Priors+Higher Order	46.6
Localization+Fusion Priors	49.2
Localization+Fusion Priors+Higher Order (small FOV)	49.3
Localization+Fusion Priors+Higher Order (large FOV)	50.9

ever, that the combination of fusion+localization gave higher accuracy than fusion only in terms of IOU. This shows the benefits of our complete multi-class masks.

Effect of the Different Components In Table 3.6, we evaluate the influence of several components of our approach. In particular, we report the results of the simple baseline mentioned above that only uses tags, but no mask. We also report the results obtained with different types of masks, with and without using the higher-order terms in our CRF smoothing procedure, and, in the multi-class case, with different network fields-of-view. The importance of our mask is clearly evidenced by the fact that mask-based results outperform the mask-free baseline by up to 17.0 mIOU points when using foreground/background masks and up to 19.9 when using multi-class masks. These results also show that using higher-order terms brings some improvement over the pairwise CRF, albeit of much lesser magnitude than the masks themselves. Similarly, the network field-of-view has some influence on accuracy. We also evaluate our approach with our additional CheckMask weak supervision procedure which yields an improvement of 4.9 and 3.5 mIOU point over our tag-only fg/bg mask approaches. However, our tag-only multi-class mask approach result are really close to the CheckMask supervision which shows the effectiveness of multi-class masks.

Computation time. For each validation image of PASCAL VOC, the network forward time is of 0.06 sec. on an NVIDIA TESLA P100 GPU. The running time of crisp boundary detection for a single image takes 4.1 seconds when using the speedy version of the public Matlab implementation [Isola et al., 2014] on a single core of an Intel Core i5 processor. For the Dense CRF, inference takes 2.8 and 2.1 seconds with and without higher-order term, respectively, using the public C++ code [Vineet et al., 2014] on a single core of an Intel Core i7 processor. The bottleneck of our approach at test time therefore is the crisp boundary detection. Note, however, that this step is only used to determine the regions for the higher-order potentials, without which, as shown in Table 3.6, our approach still yields competitive results.

Table 3.7: Per class IOU on Youtube Objects using image tags during training.

Method	bg	aero	bird	boat	car	cat	cow	dog	horse	mbike	train	mIOU
[Papazoglou and Ferrari, 2013]	-	67.4	62.5	37.8	67.0	43.5	32.7	48.9	31.3	33.1	43.4	46.8
[Tang et al., 2013]	-	17.8	19.8	22.5	38.3	23.6	26.8	23.7	14.0	12.5	40.4	23.9
[Ochs et al., 2014]	-	13.7	12.2	10.8	23.7	18.6	16.3	18.0	11.5	10.6	19.6	15.5
SEC [Kolesnikov and Lampert, 2016]	84.4	51.9	59.3	37.5	64.4	30.5	38.2	50.1	51.1	49.7	17.3	48.6
Ours (multi-class masks)	88.5	72.7	60.1	44.2	53.5	33.3	42.4	50.3	49.6	56.6	16.6	51.6

3.4.3.3 Evaluation on YTO and MS-COCO

To further demonstrate the generality of our method, we conducted a set of experiments on YTO and MS-COCO. While a few weakly-supervised methods have been applied to YTO, to the best of our knowledge, no weakly supervised results have been published on MS-COCO. We therefore also computed the results of the contemporary SEC method [Kolesnikov and Lampert, 2016] on these two datasets using the publicly available code.

Evaluation on YTO In Table 3.7 we report the per class mean IOU of our approach and several baselines on YTO. Note that our method outperforms all the baselines, including [Kolesnikov and Lampert, 2016], on this dataset.

Evaluation on MS-COCO MS-COCO is a large-scale dataset containing 80 classes of different categories. Unlike in PASCAL VOC and YTO, in MS-COCO, the majority of samples were collected from non-iconic images in a complex natural context. Moreover, a large number of the classes, e.g., spoon and knife, are *small* in terms of both size and the number of instances/samples in the datasets. Additionally, the classes of similar categories, e.g., Furniture and Indoor categories, appear together in an image, resulting in images depicting more than 10 classes. These properties make MS-COCO very challenging for weakly-supervised segmentation, and, to the best of our knowledge, we are the first to report results on this dataset in the weakly-supervised setting.

We provide the per-class IoU of our approach and SEC [Kolesnikov and Lampert, 2016] in Table 3.8. While, on average, SEC obtains slightly better results, the behavior of both methods is similar: They yield reasonable accuracy on large classes, such as Animals, but perform poorly on small ones, such as Indoor and Kitchenware. Interestingly, by analyzing the confusion matrix depicted in Figure 3.13, we noticed that our approach is more confused between classes from the same broad category. For instance, there are large confusions between the classes of the 'Food and Kitchenware' category. Furthermore, many of the classes from accessories and sport are confused with Person as in most samples they appear together with Person.

Altogether, we believe that, while promising, these results on MS-COCO evidence that there is much space for progress in weakly-supervised semantic segmentation, and, in particular, that developing solutions that improve intra-category discrimination could be an interesting direction for future research.

Table 3.8: Per class IOU on MS-COCO using image tags during training.

Cat.	Class	SEC	Ours	Cat.	Class	SEC	Ours
BG	background	74.3	68.8	Kitchenware	wine glass	22.3	17.5
P	person	43.6	27.5		cup	17.9	5.6
Vehicle	bicycle	24.2	18.2		fork	1.8	0.5
	car	15.9	7.2		knife	1.4	1.0
	motorcycle	52.1	40.5		spoon	0.6	0.6
	airplane	36.6	32.0		bowl	12.5	13.3
	bus	37.7	39.2	Food	banana	43.6	44.9
	train	30.1	26.5		apple	23.6	18.9
	truck	24.1	17.5		sandwich	22.8	21.4
	boat	17.3	16.5		orange	44.3	35.0
Outdoor	traffic light	16.7	3.9		broccoli	36.8	27.0
	fire hydrant	55.9	33.1		carrot	6.7	16.0
	stop sign	48.4	28.4		hot dog	31.2	22.5
	parking meter	25.2	25.5		pizza	50.9	57.8
	bench	16.4	12.4		donut	32.8	36.2
Animal	bird	34.7	31.1		cake	12.0	17.0
	cat	57.2	52.8	Furniture	chair	7.8	8.2
	dog	45.2	44.1		couch	5.6	13.9
	horse	34.4	34.2		potted plant	6.2	7.4
	sheep	40.3	38.0		bed	23.4	29.8
	cow	41.4	42.1		dining table	0.0	2.0
	elephant	62.9	65.2		toilet	38.5	30.1
	bear	59.1	57.0	Electronics	tv	19.2	14.8
	zebra	59.8	65.0		laptop	20.1	19.9
	giraffe	48.8	55.6		mouse	3.5	0.4
Accessory	backpack	0.3	3.2		remote	17.5	9.9
	umbrella	26.0	28.1		keyboard	12.5	19.9
	handbag	0.5	1.1		cell phone	32.1	26.1
	tie	6.5	5.5	Appliance	microwave	8.2	9.8
	suitcase	16.7	21.3		oven	13.7	16.4
Sport	frisbee	12.3	5.6		toaster	0.0	0.0
	skis	1.6	1.0		sink	10.8	9.5
	snowboard	5.3	2.8		refrigerator	4.0	13.2
	sports ball	7.9	1.9	Indoor	book	0.4	7.5
	kite	9.1	10.3		clock	17.8	16.5
	baseball bat	1.0	1.7		vase	18.4	13.4
	baseball glove	0.6	0.5		scissors	16.5	12.2
	skateboard	7.1	6.6		teddy bear	47.0	41.0
	surfboard	7.7	3.3		hair dryer	0.0	0.0
	tennis racket	9.1	5.5		toothbrush	2.8	2.0
	bottle	13.2	9.6		**mean IOU**	22.4	20.4

3.5 Conclusion

We have introduced a Deep Learning approach to weakly-supervised semantic segmentation that leverages masks directly extracted from networks pre-trained for the task of object recognition. In particular, we have shown how to extract foreground/background masks by fusing the activations of convolutional layers, as well as multi-class ones by combining this fusion-based prior with a localization one. Our experiments have shown the benefits of our masks, and in particular of the multi-class ones, which yield state-of-the-art segmentation accuracy on PASCAL VOC. A general limitation of existing tag-based semantic segmentation techniques, including ours in this chapter, is that they assume having just one background class in the scene and by relying on the object recognition pre-trained networks and objectness modules they can only segment foreground objects in the scene. However, for some real

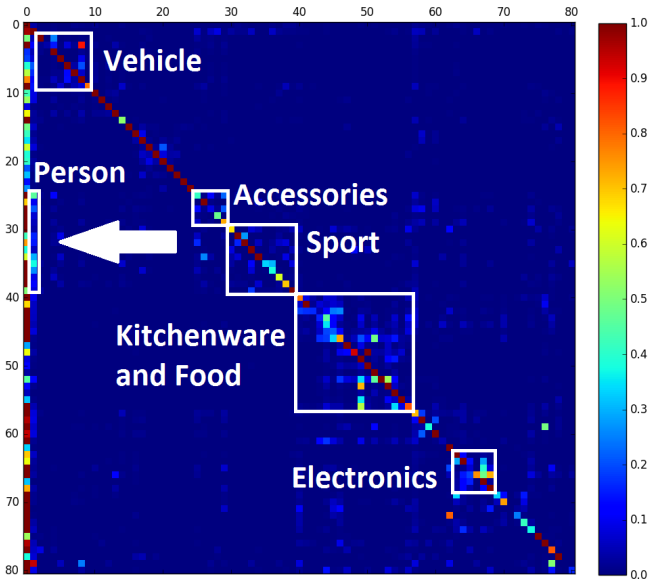


Figure 3.13: Confusion matrix of our method on the MS-COCO validation set. The classes are shown in the same order as in Table 3.8. Note that the main sources of confusion are with the background or with classes coming from the same broad category or appearing in the same context.

world applications such as autonomous navigation it is crucial to also differentiate different background classes such as road and side-walk. This is what we address in the next chapter.

Chapter 4

Making All Classes Equal in Weakly-Supervised Video Semantic Segmentation

As discussed in the previous chapter, weak supervision using only image tags could have a significant impact on semantic segmentation. Recent years have seen great progress in weakly-supervised semantic segmentation, whether from a single image or from videos. However, most existing methods are designed to handle a single background class. In practical applications, such as autonomous navigation, it is often crucial to reason about multiple background classes. In this chapter, we introduce an approach to doing so by making use of classifier heatmaps. We develop a two-stream deep architecture that jointly leverages appearance and motion, and design a loss based on our heatmaps to train it. Our experiments demonstrate the benefits of our classifier heatmaps and of our two-stream architecture on challenging urban scene datasets and on the YouTube-Objects benchmark, where we obtain state-of-the-art results.

4.1 Introduction

Video semantic segmentation, i.e., the task of assigning a semantic label to every pixel in video frames, is crucial for the success of many computer vision applications, such as video summarization and autonomous navigation. In this context, fully-supervised methods [Kundu et al., 2016; Tran et al., 2016; Shelhamer et al., 2016; Tripathi et al., 2015; Jin et al., 2016; Li et al., 2018; Jampani et al., 2017], have made great progress, particularly with the advent of deep learning. These methods, however, as we also mentioned before, inherently rely on having access to large amounts of training videos with pixel-level ground-truth annotations in every frame.

While recent years have seen great progress in weakly-supervised semantic segmentation, most existing methods, whether image- or video-based, have a major drawback: They focus on foreground object classes and treat the background as one single entity. However, having detailed information about the different background

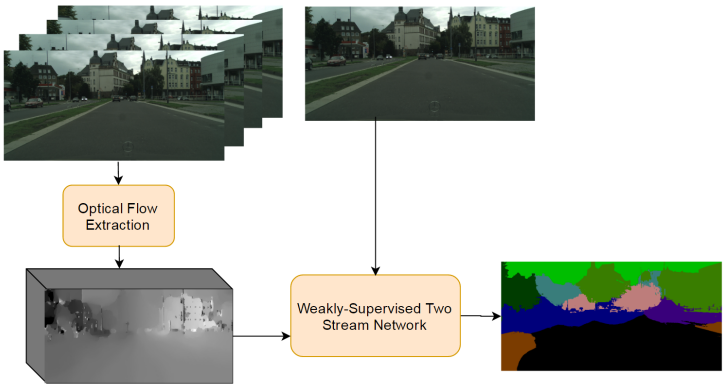


Figure 4.1: **Overview of our framework**. Given only video-level tags, our weakly-supervised video semantic segmentation network jointly leverages classifier heatmaps and motion information to model both multiple foreground classes and multiple background classes. This is in contrast with most methods that focus on foreground classes only, thus being inapplicable to scenarios where differentiating background classes is crucial, such as in autonomous driving.

classes is crucial in many practical scenarios, such as autonomous driving, where one, e.g., needs to differentiate the road from a grass field.

In this chapter, we introduce an approach to weakly-supervised video semantic segmentation that treats all classes, foreground and background ones, equally (see Fig 5.5). To this end, we propose to rely on class-dependent heatmaps obtained from classifiers trained for image-level recognition, i.e., requiring no pixel-level annotations. These classifier heatmaps provide us with valuable information about the location of instances/regions of each class. We therefore introduce a weakly-supervised loss function that let us exploit them in a deep architecture.

In particular, we develop a two-stream deep network that jointly leverages appearance and motion. Our network fuses these two complementary sources of information in two different ways: A trainable early fusion, which puts in correspondence the spatial and temporal information and learns to combine it into a spatio-temporal stream, and a late fusion further leveraging the valuable semantic information of the spatial stream to merge it with the spatio-temporal one for final prediction. Altogether, our approach constitutes the first end-to-end framework for weakly-supervised semantic segmentation to handle both multiple foreground and background classes.

To the best of our knowledge, only two weakly-supervised video semantic segmentation approaches [Liu et al., 2014; Zhong12 et al.] can potentially handle multiple background classes. However, [Liu et al., 2014] relies on a simple similarity measure between handcrafted features, and thus does not translate well to complex scenes where multiple instances of the same class have significantly different appearances. While [Zhong12 et al.] relies on more robust, pre-trained deep learning features, it exploits additional, pixel-wise annotations to train a fully-convolutional

network for scene/object classification. Furthermore, none of these two methods offer an end-to-end learning approach, which has proven key to the success of many other computer vision tasks.

Our experiments demonstrate the benefits of our approach in several scenarios. First, it yields accurate segmentations on challenging outdoor scenes, such as those depicted by the CamVid [Brostow et al., 2009a] and Cityscapes [Cordts et al., 2016] datasets, for which methods modeling foreground classes only do not apply. Furthermore, it outperforms the state-of-the-art methods that, as us, rely only on video-level tags on the standard YouTube Object [Prest et al., 2012] dataset.

4.2 Related Work

Over the years, many approaches have tackled the problem of video semantic segmentation. In particular, much research has been done in the context of fully-supervised semantic segmentation, including methods based on CNNs [Shelhamer et al., 2016; Tran et al., 2016; Jin et al., 2016; Jampani et al., 2017; Li et al., 2018] and on graphical models [Kundu et al., 2016; Liu and He, 2015; Tripathi et al., 2015]. Here, however, we focus the discussion on the methods that do not require fully-annotated training data, which is typically expensive to obtain.

In this context, semi-supervised approaches have been investigated. In particular, [Tsai et al., 2016a; Jain and Grauman, 2014a] proposed to propagate pixel-level annotations provided in the first frame of the sequence throughout the entire video. While this still requires complete annotations in one frame per video, [Shankar Nagaraja et al., 2015] relied on user scribbles to define foreground and background regions. None of these methods, however, consider background classes. Furthermore, they all still make use of some pixel-level annotations.

By contrast, weakly-supervised semantic segmentation methods tackle the challenging scenario where only weak annotations, e.g., tags, are given as labels. Much research in this context has been done for still images [Zhang et al., 2015a; Xu et al., 2014; Pourian et al., 2015; Vezhnevets et al., 2011; Pathak et al., 2015b; Vezhnevets et al., 2012; Papandreou et al., 2015; Saleh et al., 2016, 2018b; Pathak et al., 2015a; Kolesnikov and Lampert, 2016; Wei et al., 2016b,a; Shimoda and Yanai, 2016; Pinheiro and Collobert, 2015; Oh et al., 2017]. In particular, most recent methods build on deep networks by making use of objectness criteria [Bearman et al., 2016], object proposals [Pinheiro and Collobert, 2015; Wei et al., 2016a; Qi et al., 2016], saliency maps [Shimoda and Yanai, 2016; Wei et al., 2016b; Wang et al., 2018b; Oh et al., 2017], localization cues [Mostajabi et al., 2016; Kolesnikov and Lampert, 2016], convolutional activations as discussed in Chapter 3, motion cues [Tokmakov et al., 2016] and constraints related to the objects [Pathak et al., 2015a; Papandreou et al., 2015]. Since the basic networks have been pre-trained for object recognition, and thus focus on foreground classes, these methods are inherently unable to differentiate multiple background classes.

Similarly, most weakly-supervised video semantic segmentation techniques also

focus on modeling a single background class. In this context [Hartmann et al., 2012; Tang et al., 2013] work in the even more constrained scenario, where only two classes are considered: foreground vs. background. By contrast, to differentiate multiple foreground classes, but still assuming a single background, [Papazoglou and Ferrari, 2013] relied on motion cues and [Hong et al., 2017b] made use of a huge amount of web-crawled data (4606 videos with 960,517 frames).

In the same setting of multiple foreground vs. single background, several methods have proposed to rely on additional supervision. For instance, [Zhang et al., 2015b] relied on the CPMC [Carreira and Sminchisescu, 2010] region detector, which has been trained from pixel-level annotations, to segment foreground from background. In [Wang et al., 2016] and [Fragkiadaki et al., 2015], object proposal methods trained from pixel-level and bounding box annotations, respectively, were employed. Similarly, [Drayer and Brox, 2016] relied on an object detector trained from bounding boxes. The method of [Tsai et al., 2016b] utilized the FCN trained on PASCAL VOC in a fully-supervised manner to generate initial object segments.

All the weakly-supervised approaches discussed above assume to observe a single background class. In many cases, such as autonomous navigation, however, it is crucial to differentiate between multiple background classes. To the best of our knowledge, only two methods are able to handle this scenario. In [Liu et al., 2014], a nearest-neighbor-based label transfer technique was introduced, which relies on a simple distance between handcrafted features. While this strategy would work well for classes such as grass or sky in which appearance variations are limited, it translates poorly to more challenging and complex scenes, such as urban ones, where individual classes can depict a large range of appearances. As a consequence, this method was only demonstrated on simple scenes containing at most one or two instances of a few classes. In [Zhong12 et al.], more advanced, deep learning features were exploited. However, this method makes use of pixel-level supervision to train an FCN to label pixels as either scene vs. object, or multiple scene classes vs. object.

By contrast, we introduce a method that handles multiple foreground and background classes, but only relies on video-level tags. To this end, we introduce a loss function based on classifier heatmaps, and exploit it to train a two-stream network jointly leveraging complementary spatial and temporal information in an end-to-end manner.

4.3 Our Method

In this section, we introduce our approach to weakly-supervised video semantic segmentation. First, we introduce the classifier heatmaps that allow us to model both multiple foreground and background classes. We then introduce our two-stream architecture, which jointly leverages motion and appearance, and discuss our learning scheme, including our loss based on the classifier heatmaps.

4.3.1 Classifier Heatmaps

One of the main challenges when working with tags only for weakly-supervised semantic segmentation is that the annotations do not provide any information about the location of the different classes. While mitigated in the presence of only a few foreground classes and a single background one, this problem becomes highly prominent when dealing with complex urban scenes containing many instances of each foreground class and several background classes, such as road, grass, buildings. Existing weakly-supervised methods are dedicated to handle multiple foreground objects, but cannot handle multiple background ones, typically because they inherently rely on object recognition networks, which only tackle foreground classes. To address this, we propose to extract class-specific heatmaps that localize the different classes. Our goal here is to achieve this for both foreground and background classes, and without requiring any pixel-level or bounding box annotations.

Prior work has shown that ConvNets trained with a classification loss can yield remarkable localization results [Oquab et al., 2015b; Zhou et al., 2016]. Hence, similarly, for foreground classes, we make use of the VGG-16 network [Simonyan and Zisserman, 2014] trained on the standard 1000 ImageNet classes. Specifically, we transform the VGG-16 model into a fully-convolutional network by converting its fully-connected layers into convolutional ones, while keeping the trained weights. In other words, the output of the last layer of the transformed model becomes a $W \times H \times 1000$ tensor, and passing an image through the network yields a map showing the activation of each class at each pixel in a low-resolution version of the input image. In practice, we can then access the activations of the foreground classes of interest by only considering a subset of the 1000 ImageNet classes.

The standard 1000 ImageNet classes, however, do not include background. To this end, we collected iconic background images by crawling the background classes on the ImageNet website [Deng et al., 2009]. We then trained one-vs-all VGG-16 models (pre-trained on the standard 1000 ImageNet classes) for these background classes and followed the same strategy as for the foreground ones to obtain heatmaps. More details are provided in section 4.4.1.

In Figure 4.2, we show the heatmaps for some of the foreground and background classes of the Cityscapes [Cordts et al., 2016] dataset. Note that, while sometimes a bit coarse, the heatmaps still provide valuable information about the location of these classes. In the next section, we introduce our two-stream network that jointly leverages appearance and motion, and show how our heatmaps can be used to train it.

4.3.2 Weakly-supervised Two-stream Network

Videos have two intrinsic features: *Appearance* and *Motion*. To leverage these two sources of information, inspired by the approach of [Feichtenhofer et al., 2016] for action recognition, we develop the two-stream network depicted in Figure 4.3. One stream takes an RGB image as input, and the other optical flow. Compared to taking a series of images as input, explicitly using optical flow to represent motion, has the

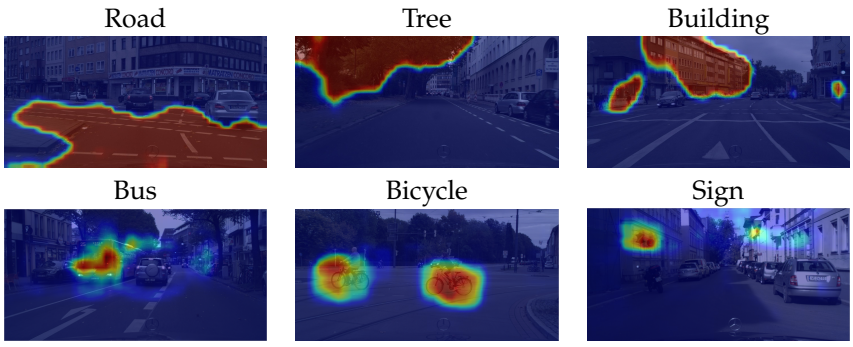


Figure 4.2: Classifier heatmaps for some of the foreground and background classes of the Cityscapes dataset. Note that these heatmaps give a good indication of the location of foreground instances and background regions.

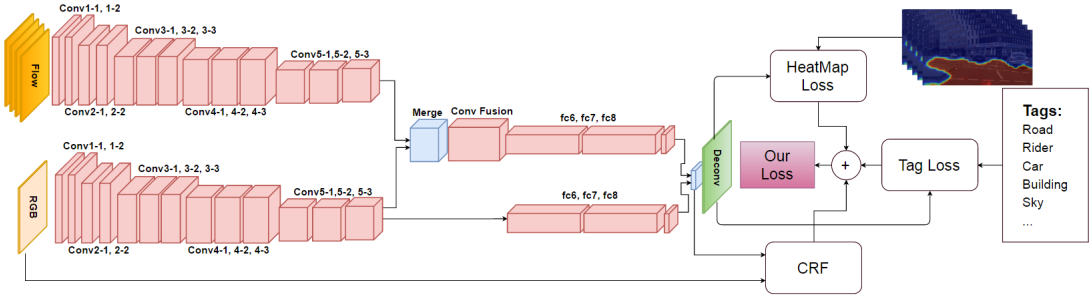


Figure 4.3: **Proposed Network Structure**. Our two-stream semantic segmentation network leverages both image and optical flow to extract the features. These features are fused in two stages. An early, trainable fusion that puts in correspondence the spatial and temporal information, and a late fusion that merges the resulting spatio-temporal stream with the appearance one for final prediction.

advantage of relieving the network from having to estimate motion implicitly. Below, we discuss how we encode optical flow and describe our fusion strategy. We then introduce our weakly-supervised learning framework.

Encoding Optical Flow. Dense optical flow [Brox et al., 2004] can be represented as a displacement vector field between a pair of frames at time t and $t+1$. The horizontal and vertical components of the displacement vector field can be thought of as image channels, which makes them well suited to act as input to a convolutional network, such as the one shown in the upper stream of the model in Figure 4.3. To represent the motion across a video clip, we stack the flow channels corresponding to both directions (vertical and horizontal) of L consecutive frames, in range $(t - \lfloor \frac{L}{2} \rfloor, t + \lfloor \frac{L}{2} \rfloor]$, to form a total of $2L$ input channels.

Fusing Appearance and Motion. As can be seen in Figure 4.3, the appearance and motion streams both consist of a series of convolutional layers, following the VGG-16 architecture [Simonyan and Zisserman, 2014]. The outputs of these streams are then fused at two different levels. In particular, fusion occurs after the fifth convolutional layer (Conv5-3) of each stream, which has been shown to contain a rich semantic representation of the input [Bertasius et al., 2014; Saleh et al., 2016, 2018b]. The first, early fusion puts in correspondence the activations of both streams corresponding to the same pixel location. As [Feichtenhofer et al., 2016], instead of performing sum- or max-fusion, we rely on a convolutional fusion strategy. This gives more flexibility to the network and allows it to learn which channels from the motion and appearance streams should be combined together. The second, late fusion of our network merges the spatio-temporal stream resulting from early fusion with the appearance stream. This fusion is achieved at the point where each stream predicts class scores. The rationale behind this is that the appearance stream provides valuable semantic information on its own, and should thus be propagated to the end of the network. The resulting scores are then passed through a deconvolution layer to obtain the final, full-resolution, semantic map.

4.3.2.1 Weakly-Supervised Learning

We now introduce our learning algorithm for weakly-supervised semantic segmentation. We first introduce a simple loss based on image tags only, and then show how we can incorporate the localization information of our classifier heatmaps to the loss function.

Intuitively, given image tags, one would like to encourage the image pixels to be labeled as one of the classes that are observed in the image, while preventing them to be assigned to unobserved classes. Note that this assumes that the full set of tags available cover all the classes depicted in the image, which is a common assumption in weakly-supervised semantic segmentation [Pathak et al., 2015a; Papandreou et al., 2015; Pathak et al., 2015b; Bearman et al., 2016; Kolesnikov and Lampert, 2016; Saleh et al., 2016, 2018b].

Formally, given an input video V, let $\mathcal{L}$ be the set of tags associated to V and $\bar{\mathcal{L}}$ the class labels that are not among the tags. Furthermore, let us denote by $s^k_{i,j}(\theta)$ the score produced by our network with parameters θ for the pixel at location (i, j) and for class k, $0 \leq k < N$, in the current input video frame I. Note that, in general, we will omit the explicit dependency of the variables on the network parameters. Finally, let $S^k_{i,j}$ be the probability of class k obtained after a softmax layer, i.e.,

$$S^k_{i,j} = \frac{\exp(s^k_{i,j})}{\sum_{c=1}^{N} \exp(s^c_{i,j})} \,. \tag{4.1}$$

Encoding the above-mentioned intuition can then simply be achieved by design-

ing a loss of the form

$$L_{tag} = -\frac{1}{|\mathcal{L}|}\sum_{k\in\mathcal{L}}\log S^k - \frac{1}{|\bar{\mathcal{L}}|}\sum_{k\in\bar{\mathcal{L}}}\log(1-S^k)\ , \tag{4.2}$$

where S^k represents a candidate score for each class in the input frame. In short, the first term in Equation 4.2 expresses the fact that the present classes should be in the input frame, while the second term penalizes the pixels that have high probabilities for the absent classes. In practice, instead of computing S^k as the maximum probability (as previously used in [Pathak et al., 2015b; Bearman et al., 2016]) for class k over all pixels in the input frame, we make use of the convex Log-Sum-Exp (LSE) approximation of the maximum (as previously used in [Pinheiro and Collobert, 2015] and in Chapter 3), which can be written as

$$\tilde{S}^k = \frac{1}{r}\log\left[\frac{1}{|I|}\sum_{i,j\in I}\exp(rS^k_{i,j})\right]\ , \tag{4.3}$$

where $|I|$ denotes the total number of pixels in the input frame and r is a parameter allowing this function to behave in a range between the maximum and the average. In practice, following our setting in Chapter 3 and [Pinheiro and Collobert, 2015], we set r to 5.

The loss of Equation 4.2 does not rely on any localization cues. As a consequence, minimizing it will typically yield poor object localization accuracy. To overcome this issue, we propose to make use of the classifier heatmaps introduced in section 4.3.1. To this end, we first generate binary masks B_k for each class k. These binary masks are obtained by setting to 1 the values that are above 20% of the maximum value in the heatmap of class k, and to 0 the other ones.

Our goal then is to encourage the model to have, for each class, high probability at pixels inside the corresponding binary mask. To this end, we introduce the loss function

$$L_{heatmap} = -\frac{1}{|\mathcal{L}|}\sum_{k\in\mathcal{L}}\frac{1}{|B_k|}\sum_{i,j\in B_k}\log S^k_{i,j}\ , \tag{4.4}$$

which we use in conjunction with the loss of Equation 4.2.

While this heatmap-based loss significantly helps localizing the different classes, the heatmaps typically only roughly match the class boundaries. To overcome this, we follow the CRF-based strategy of [Kolesnikov and Lampert, 2016]. Specifically, we construct a fully-connected CRF, with unary potentials corresponding to the probability scores predicted by our segmentation network, and image-dependent Gaussian pairwise potentials [Krähenbühl and Koltun, 2011].

Inspired by [Kolesnikov and Lampert, 2016], we then add another term to the loss function, corresponding to the mean KL-divergence between the outputs of the network and the outputs of the fully connected CRF. Specifically, we construct a fully-connected CRF, $Q(I, f(I;\theta))$, with unary potentials corresponding to the probability scores predicted by our segmentation network $f(I;\theta)$, and image-dependent

Table 4.1: Background classes used to train our classifiers (Section 4.3.1) for the Cityscapes and CamVid datasets.

Class	road	sidewalk	building	vegetation	terrain	sky
#Samples	126	306	670	176	190	180

Gaussian pairwise potentials [Krähenbühl and Koltun, 2011]. Note that the image I is downscaled, so that it matches the resolution of the segmentation mask, produced by the network.

$$L_{crf} = \frac{1}{n} \sum_{u=1}^{n} \sum_{k \in \mathcal{L}} Q_{u,k}(I, f(I;\theta)) \log \frac{Q_{u,k}(I, f(I;\theta))}{f_{u,k}(I;\theta)} \tag{4.5}$$

where $u \in 1, 2, ..., n$ is all the locations in the downscaled image and k is the classes of interest. This term encourages the network prediction to coincide with the CRF output, which produces segmentations that better respect the image boundaries.

Altogether, our network can handle multiple foreground and background classes, and, as discussed in more detail in Section 4.4.2, can be trained in an end-to-end fashion.

4.4 Experiments

In this section, we first describe the datasets used in our experiments and provide details about our learning and inference procedures. We then present the results of our model and compare it to state-of-the-art weakly-supervised semantic segmentation methods.

4.4.1 Datasets

To demonstrate the effectiveness of our approach, and evaluate the different components of our model, we use the challenging Cityscapes [Cordts et al., 2016] and CamVid [Brostow et al., 2009a] road scene datasets. Furthermore, to compare to the state-of-the-art, we make use of YouTube-Objects [Prest et al., 2012], which most weakly-supervised video semantic segmentation methods report on. Note that, although different annotation types are provided in each of these datasets, we only make use of tags, indicating which classes are present in each video clip.

Cityscapes: Cityscapes [Cordts et al., 2016] is a large-scale dataset, containing high quality pixel-level annotations for 5000 images collected in street scenes from 50 different cities. The images of Cityscapes have resolution 2048×1024, making it a challenge to train very deep networks with limited GPU memory. We therefore downsampled the images by a factor of 2.

The annotations correspond to the 20^{th} frame of 30-frame video snippets. We then extracted optical flow from 10 consecutive frames, from the 16^{th} to the 25^{th}, and

used the RGB frames and image tags in conjunction with these optical flows to train our model.

We made use of the standard training/validation/test partitions, containing 2975, 500, and 1525 images, respectively. Following the standard evaluation protocol [Cordts et al., 2016], we used 19 semantic labels (belonging to 7 super categories: ground, construction, object, nature, sky, human, and vehicle) for evaluation (the void label is not considered for evaluation).

CamVid: The CamVid dataset consists of over 10 minutes of high quality 30 Hz footage. The videos are captured at 960×720 resolution with a camera mounted inside a car. Three of the four sequences were shot in daylight, and the fourth one was captured at dusk. This dataset contains 32 categories. In our experiments, following [Brostow et al., 2009b; Kundu et al., 2016; Badrinarayanan et al., 2015], we used a subset of 11 classes. The dataset is split into 367 training, 101 validation and 233 test images. As for Cityscapes, ground-truth labels are provided every 30 frames. We extracted optical flow in 10 frames around the labeled ones, and used them with the RGB frames for training.

Iconic Data: The background classes and number of samples per class, extracted from the background images of the ImageNet website, as mentioned in Section 4.3.1, and used to train our background classifiers for Cityscapes and CamVid are given in Table 4.1. Note that, in the standard 1000 classes of ImageNet, there is no general *person* class, which appears in both datasets. To handle this class, we therefore proceeded in a similar manner as for the background classes, but making use of a small subset of the samples (1300 samples) from [Dalal and Triggs, 2005; Overett et al., 2008].

YouTube-Objects: The YouTube-Objects dataset is composed of videos collected from YouTube by querying for the names of 10 object classes of the PASCAL VOC Challenge. It contains between 9 and 24 videos per class. The duration of each video ranges from 30 seconds to 3 minutes. The videos are weakly annotated, with each video containing at least one object of the corresponding queried class. In the dataset, the videos are separated into shots.

For our experiments, we randomly extracted 6-8 frames from each shot to obtain a total of 13800 frames out of 700,000 available in the dataset. We again made use of snippets of 10 frames to encode optical flow.

For evaluation, we used the subset of images with pixel-level annotations provided by [Jain and Grauman, 2014b]. Note that there is no overlap between this subset and the shots from which we extracted the training data.

4.4.2 Implementation Details

To train our two-stream network, introduced in Section 4.3.2, we relied on stochastic gradient descent with a learning rate starting at 10^{-5} with a decrease factor of 10 every $10k$ iterations, a momentum of 0.9, a weight decay of 0.0005, and mini-batches of size 1. Similarly to recent weakly-supervised segmentation methods [Saleh et al., 2016; Bearman et al., 2016; Pathak et al., 2015b; Papandreou et al., 2015; Kolesnikov

and Lampert, 2016], the weights of our two-stream network were initialized with those of the 16-layer VGG classifier [Simonyan and Zisserman, 2014] pre-trained for 1000-way classification on the ILSVRC 2012 [Russakovsky et al., 2015]. Hence, for the last convolutional layer, we used the weights corresponding to the classes shared by the datasets used here and in ILSVRC. For the background classes, we initialized the weights with zero-mean Gaussian noise with a standard deviation of 0.1. At inference time, given only the test image and optical flow, the network generates a dense prediction as a complete semantic segmentation map.

For both CamVid and Cityscapes, we used the GPU implementation of [Brox et al., 2004] to generate the stack of optical flow for each snippet of length 10. For YouTube-Objects, we used the optical flow information pre-computed by [Brox and Malik, 2011]. Note that neither of these methods relies on any learning strategy, and thus they can be directly applied to our input images. We used C++ and Python (the Caffe framework [Jia et al., 2014]) for our implementation. As other methods [Kolesnikov and Lampert, 2016; Papandreou et al., 2015; Saleh et al., 2016, 2018b; Pathak et al., 2015a; Wei et al., 2016b], we further applied a dense CRF [Krähenbühl and Koltun, 2011] to refine this initial segmentation. To this end, we used the default CRF parameter values as in the original paper [Krähenbühl and Koltun, 2011].

4.4.3 Experimental Results

Below, we first evaluate the different components of our method on the validation set of the two challenging road scene datasets. We then provide results of our complete framework on their respective test sets. Finally, we compare our approach to state-of-the-art weakly-supervised segmentation methods on YouTube-Objects.

4.4.3.1 Ablation Study

To evaluate the influence of the different components of our approach, we designed the following baselines. *No-Heatmap* corresponds to a single-stream model exploiting the RGB image only, without exploiting our heatmap-based loss of Equation 4.4, i.e., using the loss of Equation 4.2 and the CRF loss. *Foreground-Heatmap* consists of a similar single stream network, additionally using the loss of Equation 4.4, but only for the foreground classes extracted from the VGG-16 network pre-trained on ILSVRC. *Our-Heatmap* corresponds to using all our heatmaps, i.e., for foreground and background classes, with a single-stream network. Finally, *Ours* corresponds to our two-stream network with all the loss terms.

We report the results of these different models in Table 4.2 for Cityscapes and in Table 4.3 for CamVid. In particular, we report the mean Intersection over Union (mIoU), the average per-class accuracy and global accuracy. The general behavior is the same for both datasets: Exploiting heatmaps for foreground class improves over not using heatmaps at all. However, also relying on heatmaps for background classes gives a significant boost in performance. Finally, jointly leveraging appearance and motion in our two-stream network further improves segmentation accuracy. As can

Table 4.2: Influence of our heatmaps and of optical flow. These results were obtained using the Cityscapes validation set.

Setup	Mean IOU	Mean Class Acc.	Global Acc.
No-Heatmap	8.4%	18.8%	20.9%
ImageNet-Heatmap	11.4%	33.2%	22.0%
Our-Heatmap	20.6%	40.6%	54.0%
Our Two-Stream	23.6%	40.3%	63.9%

Table 4.3: Influence of our heatmaps and of optical flow. These results were obtained using the CamVid validation set.

Setup	Mean IOU	Mean Class Acc.	Global Acc.
No-Heatmap	10.2%	24.9%	19.5%
ImageNet-Heatmap	11.0%	25.8%	28.9%
Our-Heatmap	29.5%	49.7%	62.6%
Our Two-Stream	31.1%	50.2%	67.4%

Table 4.4: Influence of our heatmaps and of optical flow. Per-class IoU for the CamVid validation set.

Setup	building	vegetation	sky	car	sign	road	pedestrian	fence	pole	sidewalk	cyclist
No-HeatMap	37.0	33.0	0.0	28.6	7.8	4.6	0.0	0.1	0.7	0.4	0
ImageNet-HeatMap	29.8	0.0	0.1	14.1	7.5	53.4	4.9	4.9	0.2	0.0	6.2
Our-HeatMap	54.1	76.1	86.3	19.4	6.6	56.3	9.0	0.9	0.5	6.0	9.0
Our Two-Stream	63.4	72.2	84.2	19.3	8.9	60.6	14.3	0.0	0.0	4.1	15.2

Table 4.5: Comparison to fully-supervised semantic segmentation methods on the CamVid test set. While we use the weakest level of supervision, the difference to fully supervised methods, especially in background classes (sky, building, road and tree) is remarkably low.

Method	Supervision	building	vegetation	sky	car	sign	road	pedestrian	fence	pole	sidewalk	cyclist	mIOU
SegNet	pixel level	68.7	52.0	87.0	58.5	13.4	86.2	25.3	17.9	16.0	60.5	24.8	46.4
[Liu and He, 2015]	pixel level	66.8	66.6	90.1	62.9	21.4	85.8	28.0	17.8	8.3	63.5	8.5	47.2
Ours	image tags	58.9	46.4	83.8	26.5	12.0	64.4	8.0	11.3	3.1	1.1	11.0	29.7

be observed in Table 4.4, which provides the per-class intersection over union for CamVid, our heatmaps and our two-stream network add significant improvement to the baselines for most of the classes, especially in background classes, e.g., sky and road.

Furthermore, we evaluated the influence of the CRF on our results. On Cityscapes, our two-stream network without the CRF loss achieves 20.3% mIOU vs 23.6% with the CRF, thus showing that the CRF helps, but is not the key to our results.

Regarding runtimes, the average inference time of our method per image on

Table 4.6: Comparison to fully-supervised semantic segmentation methods on the Cityscapes test set. As on CamVid, while we use the weakest level of supervision, the gap with fully supervised methods is quite low, particularly on background classes.

Method	Supervision	road	sidewalk	building	wall	fence	pole	traffic-light	traffic-sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIOU Class	mIOU Category
FCN-8s	pixel-level	97.4	78.4	89.2	34.9	44.2	47.4	60.1	65.0	91.4	69.3	93.9	77.1	51.4	92.6	35.3	48.6	46.5	51.6	66.8	65.3	85.7
Deeplab	pixel-level	97.3	77.6	87.7	43.6	40.4	29.7	44.5	55.4	89.4	67.0	92.7	71.2	49.4	91.4	48.7	56.7	49.1	47.9	58.6	63.1	81.2
SegNet	pixel-level	96.4	73.2	84.0	28.4	29.0	35.7	39.8	45.1	87.0	63.8	91.8	62.8	42.8	89.3	38.1	43.1	44.1	35.8	51.9	56.9	79.1
Ours	image tags	78.5	2.7	45.0	6.6	9.8	5.4	0.7	2.1	63.3	22.0	71.5	17.6	8.0	43.6	16.0	15.5	33.0	17.9	13.6	24.9	47.2

Table 4.7: Comparison to the state-of-the-art on the YouTube-Objects dataset. We report the per-class and mean IoU. Note that our two-stream network significantly outperforms the state-of-the-art baselines.

Method	aeroplane	bird	boat	car	cat	cow	dog	horse	motorbike	train	mIOU
[Papazoglou and Ferrari, 2013]	67.4	62.5	37.8	67.0	43.5	32.7	48.9	31.3	33.1	43.4	46.8
[Tang et al., 2013]	17.8	19.8	22.5	38.3	23.6	26.8	23.7	14.0	12.5	40.4	23.9
[Ochs et al., 2014]	13.7	12.2	10.8	23.7	18.6	16.3	18.0	11.5	10.6	19.6	15.5
Ours	67.6	72.3	58.1	60.1	59.8	42.6	60.1	46.3	53.6	12.4	53.3

Cityscapes given optical flow is 0.56s without CRF inference as post-processing and 3.6s with CRF inference. This matches the runtimes reported in other papers that worked on Cityscapes, although in the fully-supervised setting, such as [Long et al., 2015] (0.5s without CRF) and [Chen et al., 2014] (4s with CRF).

4.4.3.2 Results on Test Sets

We then evaluated our complete approach on the test sets of CamVid and Cityscapes. In Table 4.5 and Table 4.6, we compare the results of our weakly-supervised approach to those of fully-supervised methods. Note that, while these methods make use of much stronger supervision during training, thus making the comparison unfair to us, the gap in accuracy with our method, especially for background classes (sky, building, road and tree) is remarkably low. This further illustrates the strength of our approach, which, despite using only tags, yields good segmentation accuracy.

Qualitative results of our two-stream network on samples from Cityscapes and CamVid are also depicted in Figure 4.4.

4.4.3.3 Comparison to the State-of-the-art

To further show the effectiveness of our method, we compare it with other weakly-supervised video semantic segmentation baselines on the standard YouTube-Objects dataset. Note that, here, all the classes correspond to foreground objects, with a single background class, which makes this dataset a less attractive candidate for

our method. This comparison, however, lets us evaluate the performance of our two-stream network with respect to the state-of-the-art in weakly-supervised video semantic segmentation. As shown in Table 4.7, our results significantly outperform the state-of-the-art on this dataset, thus again showing the benefits of our approach (see Figure 4.4 for qualitative results).

Note that other approaches that make use of additional supervision, such as object detectors trained from pixel-level [Zhang et al., 2015b] or bounding box [Drayer and Brox, 2016] annotations, have also reported results on this dataset. While we only exploit tags, our approach yields results comparable to those of these methods (53.3% for our method versus 54.1% for [Zhang et al., 2015b] and 55.8% for [Drayer and Brox, 2016]).

4.5 Conclusion

In this chapter, we have proposed the first weakly-supervised video semantic segmentation approach that considers both multiple foreground and background classes. To this end, we have introduced a two-stream network that leverages both optical-flow and RGB images, trained using a loss based on classifier heatmaps. Our experiments demonstrated the benefits of using such heatmaps and of exploiting optical flow on challenging urban datasets. Furthermore, our two-stream network has also outperformed the state-of-the-art weakly-supervised video semantic segmentation methods on the standard YouTube-Object benchmark. Although moving one step towards weakly-supervised semantic segmentation in the scenarios where we need to segment multiple background classes is valuable, the performance gap (especially in the foreground classes) compared to the fully-supervised setting is considerable. Recently, using synthetic data with automatically obtained annotations has also gained a lot of attention. In the next chapter, we focus on using synthetic data in an effective way so as to decrease the effort of manual labeling to its minimum level, while, increasing the performance of urban scene semantic segmentation considerably.

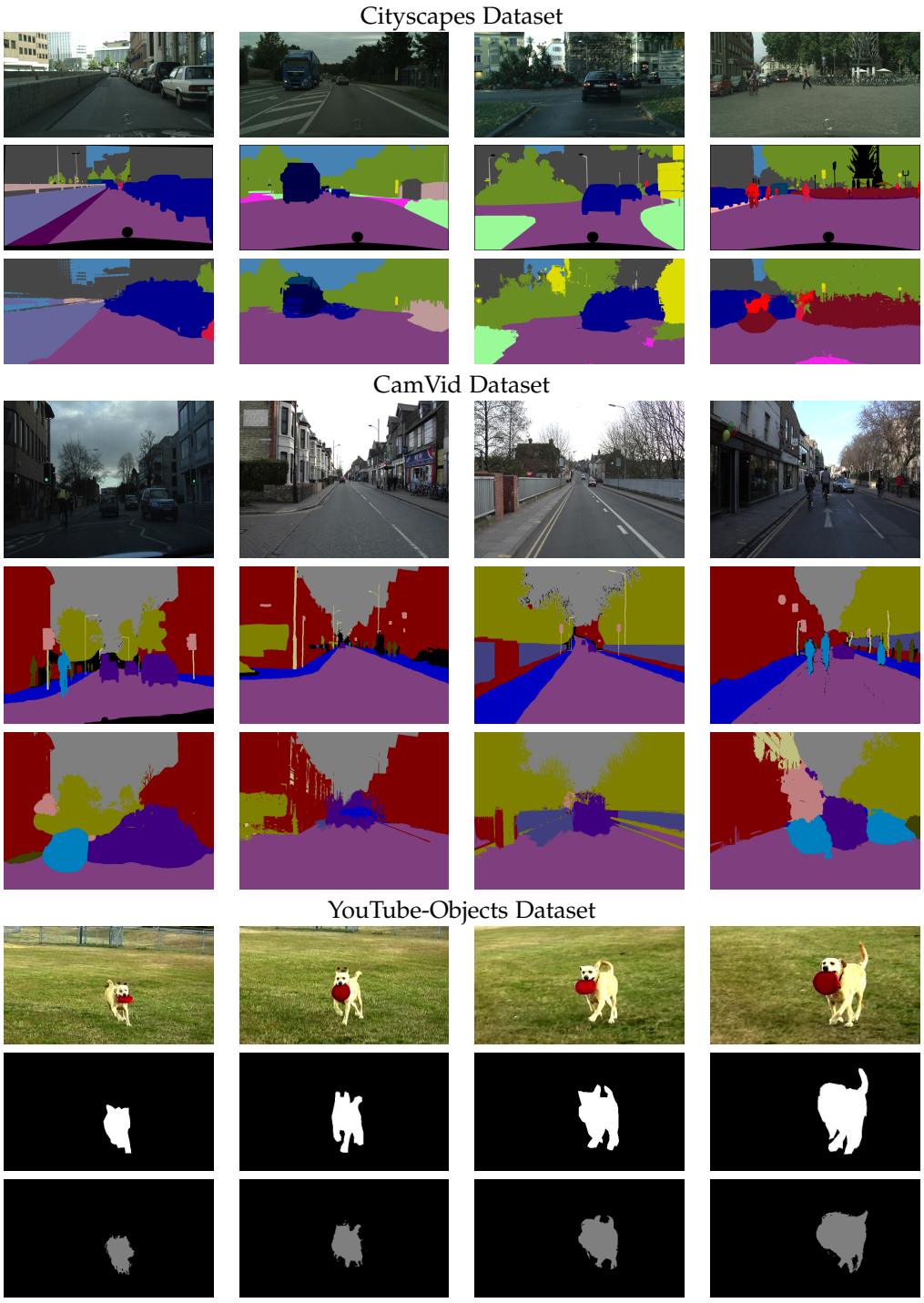


Figure 4.4: Qualitative results on Cityscapes, CamVid, and YouTube-Objects. Note that for each dataset, from top to bottom, there is the RGB frame, Ground-truth and the prediction of our two-stream network.

Chapter 5

Effective Use of Synthetic Data for Urban Scene Semantic Segmentation

As discussed in the previous chapters, training a deep network to perform semantic segmentation in a fully-supervised setting requires large amounts of labeled data. To alleviate the manual effort of annotating real images, researchers have investigated the use of synthetic data, which can be labeled automatically. Unfortunately, a network trained on synthetic data performs relatively poorly on real images. While this can be addressed by domain adaptation, existing methods all require having access to real images during training. In this chapter, we introduce a drastically different way to handle synthetic images that does not require seeing any real images at training time. Our approach builds on the observation that foreground and background classes are not affected in the same manner by the domain shift, and thus should be treated differently. In particular, foreground classes which their shape looks more natural that their texture in synthetic domain, should be handled in a detection-based manner. Our experiments evidence the effectiveness of our approach on Cityscapes and CamVid with models trained on synthetic data only.

5.1 Introduction

With the growing advance in computer graphics, using synthetic data with automatically annotated data for different computer vision tasks and in particular, semantic scene segmentation, has obtained lots of interest in recent years. However, despite the increasing realism of synthetic data, there remain significant perceptual differences between synthetic and real images. Therefore, the performance of a state-of-the-art semantic segmentation network, such as [Chen et al., 2014; Long et al., 2015; Zhao et al., 2017; Noh et al., 2015], trained on synthetic data and tested on real images remains disappointingly low. While domain adaptation methods [Chen et al., 2017a; Hoffman et al., 2017, 2016; Zhang et al., 2017; Murez et al., 2017; Chen et al., 2017b] can improve such performance by explicitly accounting for the domain shift between

Methods	Traffic light	Traffic sign	Person	Rider	Car	Truck	Bus	Train	Motor-cycle	Bicycle
Segmentation	22.3	23.8	48.7	13.3	75.1	14.3	21.2	2.1	24.2	7.3
Detection-based	26.7	42.5	52.2	28.5	76.2	19.6	31.6	6.9	18.1	9.8

Figure 5.1: Visual comparison of different classes in real Cityscapes images (Top) and synthetic GTA5 ones (Middle). Background classes (first 4 columns) are much less affected by the domain shift than foreground ones (last 3 columns), which present clearly noticeable differences in texture, but whose shape remain realistic. (Bottom) We compare the accuracy of a semantic segmentation network (DeepLab) and of a detection-based model (Mask R-CNN), both trained on synthetic data only, on the foreground classes of Cityscapes. Note that the detection-based approach, by leveraging shape, yields significantly better results than the segmentation one.

real and synthetic data, they require having access to a large set of real images, albeit unsupervised, during training. As such, one cannot simply deploy a model trained off-line on synthetic data in a new, real-world environment.

In this chapter, we introduce a drastically different approach to addressing the mismatch between real and synthetic data, based on the following observation: Not all classes suffer from the same type and degree of perceptual differences. In particular, as can be seen in Figure 5.1, the *texture* of background classes in synthetic images looks more realistic than that of foreground classes[1]. Nevertheless, the *shape* of foreground objects in synthetic images looks very natural. We therefore argue that these two different kinds of classes should be treated differently. Specifically, we argue that semantic segmentation networks are well-suited to handle background classes because of their texture realism. By contrast, we expect object detectors to be more appropriate for foreground classes, particularly considering that modern detectors rely on generic object proposals. Indeed, when dealing with all possible texture variations of all foreground object classes, the main source of information to discriminate a foreground object from the background is shape.

[1]We distinguish foreground classes from background ones primarily based on whether they have a well-defined shape and come in instances, or they are shapeless and identified by texture or material property. In essence, this corresponds to the distinction between *things* and *stuff* in [Heitz and Koller, 2008]. See Figure 5.1 for examples.

Figure 5.2: Aerial views of our synthetic VEIS environment.

To empirically sustain our claim that detectors are better-suited for foreground classes, we trained separately a state-of-the-art DeepLab [Chen et al., 2014] semantic segmentation network and a Mask R-CNN [He et al., 2017], which performs object detection followed by binary segmentation and class prediction, on synthetic data. At the bottom of Figure 5.1, we compare the mean Intersection over Union (mIoU) of these two models on the foreground classes of Cityscapes. Note that, except for *motorcycle*, the detector-based approach outperforms the semantic segmentation network on all classes.

Motivated by this observation, we therefore develop a simple, yet effective semantic segmentation framework that better leverages synthetic data during training. In essence, our model combines the foreground masks produced by Mask R-CNN with the pixel-wise predictions of the DeepLab semantic segmentation network. Our experiments on Cityscapes [Cordts et al., 2016] and CamVid [Brostow et al., 2009a] demonstrate that this yields significantly higher segmentation accuracies on real data than if only training a semantic segmentation network on synthetic data. Furthermore, our approach outperforms the state-of-the-art domain adaptation techniques [Chen et al., 2017a; Hoffman et al., 2017, 2016; Zhang et al., 2017] without having seen any real images during training, and can be further improved by making use of unsupervised real images.

Furthermore, as a secondary contribution, we introduce a virtual environment created in the Unity3D framework, called **VEIS** (Virtual Environment for Instance Segmentation). This was motivated by the fact that existing synthetic datasets [Ros et al., 2016; Richter et al., 2016, 2017] do not provide instance-level segmentation annotations for all the foreground classes of standard real datasets, such as Cityscapes. VEIS automatically annotates synthetic images with instance-level segmentation for foreground classes. It captures urban scenes, such as those in Figure 5.2 shown from an aerial view, using a virtual camera mounted on a virtual car, yielding images such as those in Figure 5.6.

While not highly realistic, we show that, when used with a detector-based approach, this data allows us to boost semantic segmentation performance, despite it being of only little use in a standard semantic segmentation framework. The VEIS dataset is available at `https://github.com/fatemehSLH/VEIS`.

5.2 Related Work

Semantic segmentation, that is, understanding an image at pixel-level, has been widely studied by the computer vision community [Shotton et al., 2006; Tighe and Lazebnik, 2010; Gould et al., 2008; Mottaghi et al., 2014; Farabet et al., 2013; Pinheiro and Collobert, 2014; Sharma et al., 2015; Long et al., 2015; Noh et al., 2015; Chen et al., 2014; Zheng et al., 2015; Zhao et al., 2017]. As for many other tasks, the most recent techniques rely on deep networks [Chen et al., 2014; Long et al., 2015; Zhao et al., 2017; Noh et al., 2015]. Unfortunately, in contrast to image recognition problems, obtaining fully-supervised data for semantic segmentation, with pixel-level annotations, is very expensive and time-consuming. Two trends have therefore been investigated to overcome this limitation: Weakly-supervised methods and the use of synthetic data.

As also discussed in previous chapters, weakly-supervised semantic segmentation aims to leverage a weaker form of annotation which are cheaper to obtain. While great progress has been made in this area, most existing methods focus only on foreground object classes and treat the background as one single entity. However, having detailed information about the different background classes is crucial in many practical scenarios, such as automated driving, where one needs to differentiate, e.g., the road from a grass field. To the best of our knowledge, our contribution in Chapter 4 [Saleh et al., 2017] constitutes the only method that considers multiple background classes for weakly-supervised semantic segmentation. This is achieved by leveraging both appearance and motion via a two-stream architecture trained using a loss based on classifier heatmaps. While this method is reasonably effective at segmenting background classes, there is still a huge gap compared to fully-supervised methods, especially in the foreground classes.

With the advance of computer graphics, generating fully-supervised synthetic data has become an attractive alternative to weakly-supervised learning. This has led to several datasets, such as SYNTHIA [Ros et al., 2016], GTA5 [Richter et al., 2016] and VIPER [Richter et al., 2017], as well as virtual environments to generate data [Dosovitskiy et al., 2017]. Unfortunately, despite the growing realism of such synthetic data, simply training a deep network on synthetic images to apply it to real ones still yields disappointing results. This problem is due to the domain shift between real and synthetic data, and has thus been tackled by domain adaptation methods [Chen et al., 2017a; Hoffman et al., 2017, 2016; Zhang et al., 2017; Murez et al., 2017; Chen et al., 2017b], which, in essence, aim to reduce the gap between the feature distributions of the two domains. In [Hoffman et al., 2016], this is achieved by a domain adversarial training strategy inspired by the method of [Ganin and Lempitsky, 2015; Ganin et al., 2016]. This is further extended in [Chen et al., 2017b] to align not only global, but also class-specific statistics. Domain adversarial training is combined in [Chen et al., 2017a] with a feature regularizer based on the notion of distillation [Hinton et al., 2015]. In [Zhang et al., 2017], a curriculum style learning is introduced to align the label distribution over both entire images and superpixels. By contrast, [Hoffman et al., 2017] and [Murez et al., 2017] rely on a generative approach

with cycle consistency to adapt the pixel-level and feature-level representations.

While these methods outperform simply training a network on the synthetic data, without any form of adaptation, they all rely on having access to real images, without supervision, during training. As such, they cannot be directly deployed in a new environment without undergoing a new training phase.

Here, we follow an orthogonal approach to leverage synthetic data, based on the observation that foreground and background classes are subject to different perceptual mismatches between synthetic and real images. We therefore propose to rely on a standard semantic segmentation network for background classes, whose textures look quite realistic, and on a detection-based strategy for foreground objects because, while their textures look less natural, their shapes are realistic. Our experiments evidence that this outperforms state-of-the-art domain adaptation strategies. However, being orthogonal to domain adaptation, our method could also be used in conjunction with domain adaptation techniques. As a matter of fact, [Sun and Saenko, 2014], which also argues that modern detectors rely on shape and discard the background texture, introduces a domain adaptation approach for the task of object detection, which could potentially be leveraged to deal with the foreground classes in our approach. This, however, goes beyond the scope of this work.

5.3 Our Method

In this section, we introduce our approach to effectively use synthetic data for semantic segmentation in real driving scenarios. Note that, while we focus on driving scenarios, our approach generalizes to other semantic segmentation problems. However, synthetic data is typically easier to generate for urban scenes. Below, we first consider the case where we do not have access to any real images during training. We then introduce a simple strategy to leverage the availability of unsupervised real images.

5.3.1 Detection-based Semantic Segmentation

As discussed above, and illustrated by Figure 5.1, the perceptual differences of foreground and background classes in synthetic and real images are different. In fact, background classes in synthetic images look quite realistic, presenting very natural textures, whereas the texture of foreground classes does look synthetic, but their shape is realistic. We therefore propose to handle the background classes with a semantic segmentation network, but rather make use of a detection-based technique for the foreground classes. Below, we describe this in more detail, and then discuss how we perform semantic segmentation on a real image.

5.3.1.1 Dealing with Background Classes

To handle the background classes, we make use of the VGG16-based DeepLab model, depicted in Figure 5.3. Specifically, we use DeepLab with a large field of view and

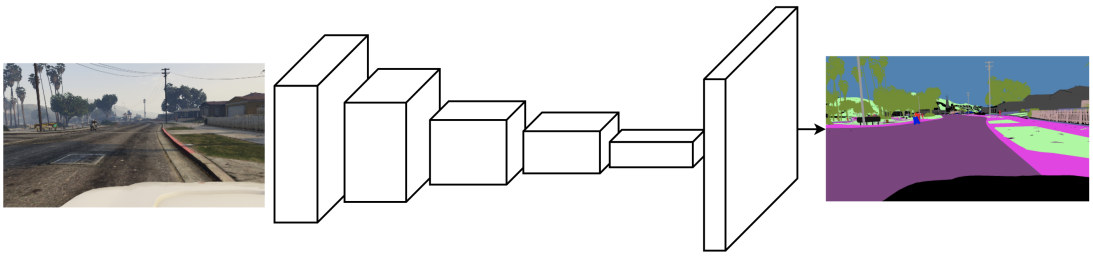


Figure 5.3: **Dealing with background classes.** We make use of the DeepLab semantic segmentation framework trained on synthetic GTA5 [Richter et al., 2016] frames with corresponding per-pixel annotations.

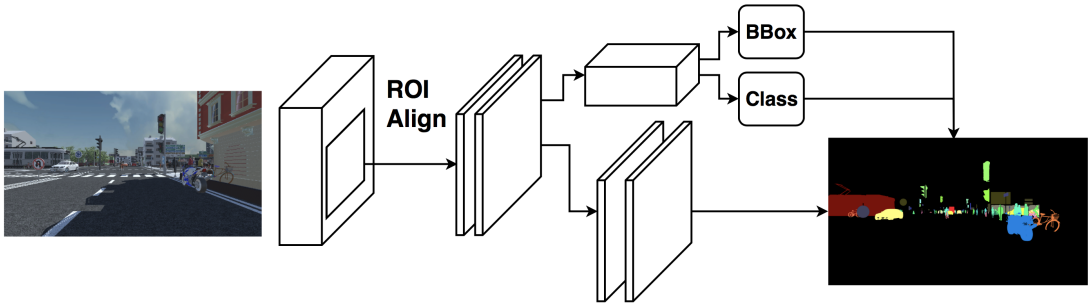


Figure 5.4: **Dealing with foreground classes.** We rely on the detection-based Mask R-CNN framework trained on our synthetic VEIS data with instance-level annotations. Note that these annotations were obtained automatically.

dilated convolution layers [Chen et al., 2014]. We train this model on the GTA5 dataset [Richter et al., 2016] in which the background classes look photo-realistic. The choice of this dataset above others was also motivated by the fact that it contains all the classes of the commonly used real datasets, such as Cityscapes and CamVid. To train our model, we use the cross-entropy loss between the network's predictions and the ground-truth pixel-wise annotations of the synthetic images. Note that the network is trained on all classes, both foreground and background, but, as explained later, the foreground predictions are mostly discarded by our approach.

5.3.1.2 Dealing with Foreground Classes

For foreground classes, our goal is to make use of a detection-based approach, which, as argued in Section 5.1, relies more strongly on object shape than on texture, thus making texture realism of the synthetic data less crucial. Since our final goal is to produce a pixel-wise segmentation of the objects, we propose to rely on a detection-based instance-level semantic segmentation technique. Note that, once an object has been detected, segmenting it from the background within its bounding box is a comparatively easier task than semantic segmentation of an entire image. Therefore,

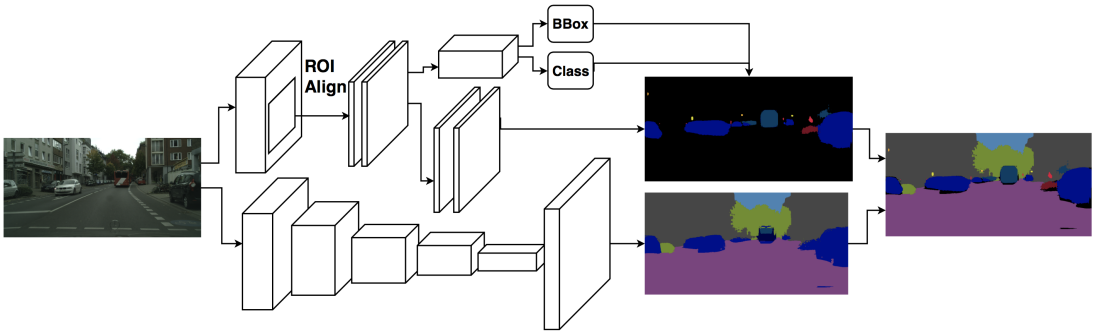


Figure 5.5: **Fusing foreground and background predictions.** Our approach combines the detection-based foreground predictions with the results of the semantic segmentation approach. Note that we do *not* require seeing any real images during training.

texture realism is also not crucial here. To address this task, we make use of Mask R-CNN [He et al., 2017], which satisfies our criteria: As illustrated in Figure 5.4, it relies on an initial object detection stage, followed by a binary mask extraction together with object classification. Since existing synthetic datasets do not provide instance-level segmentations for all foreground classes of standard real datasets, we train Mask R-CNN using our own synthetic data, discussed in Section 5.4. We make use of the standard architecture described in [He et al., 2017], as well as of the standard loss, which combines detection, segmentation, classification and regression terms.

5.3.1.3 Prediction on Real Images

The two networks described above are trained using synthetic data only. At test time, we can then feed a real image to each network to obtain predictions. However, our goal is to obtain a single, pixel-wise semantic segmentation, not two separate kinds of outputs. To achieve this, as illustrated in Figure 5.5, we fuse the two kinds of predictions, starting from the Mask R-CNN ones.

Specifically, given the Mask R-CNN predictions, we follow a strategy inspired by the panoptic segmentation procedure of [Kirillov et al., 2018], which constitutes an NMS-like approach to combine instance segments. More precisely, we first sort the predicted segments according to their confidence scores, and then iterate over this sorted list, starting from the most confident segment. If the current segment candidate overlaps with a previous segment, we remove the pixels in the overlapping region. The original procedure of [Kirillov et al., 2018] relies on two different thresholds: One to discard the low-scoring segments and the other to discard non-overlapping yet too small segment regions. The values of these thresholds were obtained by grid search on real images. Since we do not have access to the ground-truth annotations of the real images, and in fact not even access to the real images during training, we ignore these two heuristics to discard segments, and thus con-

sider all segments and all non-overlapping segment regions when combining the Mask R-CNN predictions.

Combining the Mask R-CNN predictions yield a semantic segmentation map that only contains foreground classes and has a large number of holes, where no foreground objects were found. To obtain our final semantic segmentation map, we fill these holes with the predictions obtained by the DeepLab network. That is, every pixel that is not already assigned to a foreground class takes the label with the highest probability at that pixel location in the DeepLab result.

5.3.2 Leveraging Unsupervised Real Images

The method described in Section 5.3.1 uses only synthetic images during training. In some scenarios, however, it is possible to have access to unlabeled real images at training time. This is in fact the assumption made by domain adaptation techniques. To extend our approach to this scenario, we propose to treat the predictions obtained by the method of Section 5.3.1 as pseudo ground-truth labels for the real images. To be precise, we make a small change to these predictions: In the holes left after combining the Mask R-CNN predictions, we assign the pixels that are predicted as foreground classes by the DeepLab model to an *ignore* label, so that they are not used for training. This is motivated by the fact that, as discussed above, the predictions of foreground classes by a standard semantic segmentation network are not reliable. We then use the resulting pseudo-labels as ground-truth to train a DeepLab semantic segmentation network from real images. As will be shown in our results, thanks to the good quality of our initial predictions, this helps further boost segmentation accuracy.

5.4 The VEIS Environment and Dataset

In this section, we introduce our Virtual Environment for Instance Segmentation (VEIS) and the resulting dataset used in our experiments. While there are already a number of synthetic datasets for the task of semantic segmentation in urban scenes [Ros et al., 2016; Richter et al., 2016, 2017], they each suffer from some drawbacks. In particular, GTA5 [Richter et al., 2016] does not have instance-level annotations, and is thus not suitable for our purpose. By contrast, SYNTHIA [Ros et al., 2016] and VIPER [Richter et al., 2017] do have instance-level annotations, but not for all foreground classes of commonly-used real datasets, such as Cityscapes. For instance, *train*, *truck*, *traffic light* and *traffic sign* are missing in SYNTHIA, and *rider*, *traffic sign*, *train* and *bicycle* in VIPER. Furthermore, [Richter et al., 2016, 2017] were acquired using the commercial game engine Grand Theft Auto V (GTAV), which only provides limited freedom for customization and control over the scenes and objects to be captured, thus making it difficult to obtain a large diversity and a good balance of classes. Obtaining ground-truth instance-level annotations in the GTAV game also involves a rather complicated procedure [Richter et al., 2017].

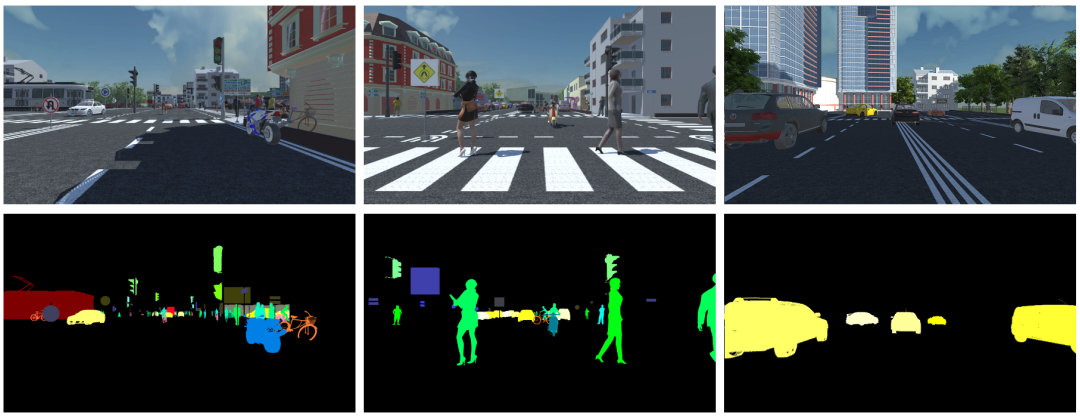


Figure 5.6: Example images and corresponding instance-level annotations, obtained automatically, from our synthetic VEIS dataset.

5.4.1 Environment

To alleviate these difficulties, we used the Unity3D [Unity3D] game engine, in which one can manually design scenes with common urban structures and add freely-available 3D objects, representing foreground classes, to the scene. Example 3D scenes are shown in Figure 5.2. Having access to the source code and manually constructing the scenes both facilitate generating annotations such as instance-level pixel-wise labels automatically. Specifically, before starting to generate the frames, our framework counts the number of instances of each class, and then assigns a unique ID to each instance. These unique IDs then automatically create unique textures and shaders for their corresponding instances. When data generation starts, both the original textures and shaders and the automatically created ones are rendered, thus allowing us to capture the synthetic image and the instance-level semantic segmentation map at the same time and in real time. Creating VEIS took 1 day for 1 person. This is very little effort, considering that VEIS allows us to have access to a virtually unlimited number of annotated images with the object classes of standard real urban scene datasets, such as CamVid and Cityscapes.

As can be seen from the samples shown in Figure 5.6, the images generated by VEIS look less photo-realistic than those of [Richter et al., 2016, 2017]. Therefore, as evidenced by our experiments, using them to train a semantic segmentation network does not significantly help improve accuracy on real images compared to using existing synthetic datasets. However, using these images within our proposed detection-based framework allows us to significantly improve semantic segmentation quality. This is due to the fact that, while not realistic in texture, the foreground objects generated by VEIS are realistic in shape, and our environment allowed us to cover a wide range of shape and pose variations.

Note that, in principle, we could have used other open source frameworks to generate our data, such as CARLA [Dosovitskiy et al., 2017], implemented as an

open-source layer of the Unreal Engine 4 (UE4) [Epic-Games]. However, CARLA is somewhat too advanced for the purpose of our investigation. It targets the complete autonomous driving pipeline, with three different approaches covering a standard modular pipeline, an end-to-end approach based on imitation learning, and an end-to-end approach based on reinforcement learning. Since our goal was only to generate synthetic images covering a large diversity of foreground objects, we found Unity3D to be sufficient and easier to deploy.

5.4.2 The VEIS Dataset

Using our VEIS environment, we generated images from two different types of scenes: 1) A multi-class, complex scene, where a city-like environment was synthesized with various objects of different classes. 2) A single-class, simple scene, where one or multiple objects of a single class were placed in a single road with background items (e.g., road, sidewalk, building, tree, sky), and images from multiple views were captured. Our VEIS dataset then contains 30180 frames from the multi-class scene and 31125 frames from the single-class scene, amounting to a total of 61305 frames with corresponding instance-level semantic segmentation. Note that the instance-level annotations were obtained with no human intervention. Some statistics of this dataset are shown in Table 5.1. In particular, we used a small amount of unique 3D objects for most of the classes and just repeated them in the scenes but with varying pose and articulation where applicable.

5.5 Experiments

In this section, we first describe the datasets used in our experiments and provide details about our learning and inference procedures. We then present the results of our model and compare it to state-of-the-art weakly-supervised semantic segmentation and domain adaptation methods.

5.5.1 Datasets

To train our model and the baseline, we make use of the synthetic GTA5 dataset [Richter et al., 2016] and of our new VEIS dataset introduced in Section 5.4. Furthermore, we also provide results of fully-supervised models trained on the synthetic SYNTHIA [Ros et al., 2016] and VIPER [Richter et al., 2017] datasets. At test time, we evaluate the models on the real images of the Cityscapes [Cordts et al., 2016] and CamVid [Brostow et al., 2009a] road scene datasets. Below, we briefly discuss the characteristics of these datasets.

GTA5 [Richter et al., 2016] was captured using the Grand Theft Auto V video game and contains 24966 photo-realistic images with corresponding pixel-level annotations. The resolution of the images is 1920×1080 and the class definitions of the semantic categories are compatible with those in the Cityscapes dataset.

Table 5.1: Some statistics of our synthetic data

Class	*T. light*	*T. sign*	*Person*	*Rider*	*Car*	*Truck*	*Bus*	*Train*	*M. bike*	*Bicycle*
#unique instances	3	69	31	1	13	6	3	3	7	4
#instances in Dataset	101771	261015	176552	67073	148760	26847	45082	12071	50687	67672

VIPER [Richter et al., 2017] is a slightly more recent dataset than GTA5, also acquired using the Grand Theft Auto V video game, but covering a wider range of weather conditions. It contains more than 250K high-resolution (1920×1080) video frames, all annotated with ground-truth labels for both low-level and high-level vision tasks, including optical flow, semantic instance segmentation, object detection and tracking, object-level 3D scene layout, and visual odometry. In our experiments, the model exploiting VIPER was trained using the training and validation sets of this dataset (over 180K frames). While VIPER is larger than GTA5, its labels are not really compatible with Cityscapes. For example, the classes *rider* and *wall* are missing; the class *pole* has been incorporated into *infrastructure*; the windows of the cars are not labeled as *car* unlike in Cityscapes. This explains why most of our experiments rather rely on the GTA5 dataset.

SYNTHIA [Ros et al., 2016] is another dataset of synthetic images, with a subset called SYNTHIA-RAND-CITYSCAPES meant to be compatible with Cityscapes. This subset contains 9,400 images with pixel-level semantic annotations. However, some classes, such as *train*, *truck* and *terrain*, have no annotations. As for VIPER, we show the performance of a fully-supervised method trained on SYNTHIA. This is for the sake of completeness, even though we favor GTA5 since it contains all the classes of Cityscapes.

Cityscapes [Cordts et al., 2016] is a large-scale dataset of real images, containing high-quality pixel-level annotations for 5000 images collected in street scenes from 50 different cities. There is also another set of images with coarse level annotations. We report the results of all models on the 500 validation images. Furthermore, the methods that rely on unsupervised real images during training, including our approach of Section 5.3.2, were trained using the 22971 train/train-extra RGB frames of this dataset.

CamVid [Brostow et al., 2009a] consists of over 10 minutes of high quality 30 Hz footage. The videos were captured at 960 × 720 resolution with a camera mounted inside a car. Three of the four sequences were shot in daylight, and the fourth one was captured at dusk. This dataset contains 32 categories. In our experiments, following [Brostow et al., 2009a], we used a subset of 11 classes. The dataset is split into 367 training, 101 validation and 233 test images. Note that, as for the Cityscapes dataset, we evaluate on the test set and, when training on unsupervised data, used the RGB frames of training+validation without any type of annotation.

5.5.2 Implementation Details

As discussed in Section 5.3, our approach makes use of two types of networks: DeepLab [Chen et al., 2014] for semantic segmentation and Mask R-CNN [He et al., 2017] for instance-level segmentation. Below, we briefly discuss these models.

5.5.2.1 DeepLab

To train our semantic segmentation networks, using either the synthetic datasets or real images with pseudo ground truth, we used a DeepLab model with a large field of view and dilated convolution layers. We relied on stochastic gradient descent with a learning rate starting at 25×10^{-5}, with a decrease factor of 10 every $40k$ iterations, a momentum of 0.9, a weight decay of 0.0005, and mini-batches of size 1. Similarly to recent semantic segmentation methods [Saleh et al., 2017; Chen et al., 2014; Long et al., 2015; Zhao et al., 2017], the weights of our semantic segmentation network were initialized with those of the 16-layer VGG classifier [Simonyan and Zisserman, 2014] pre-trained on ImageNet [Russakovsky et al., 2015]. Note that, because of limited GPU memory, we down-sampled the high resolution images of Cityscapes, GTA5, VIPER, and SYNTHIA by a factor 2 when using them for training.

5.5.2.2 Mask R-CNN

To train a Mask R-CNN network, we make use of the implementation provided by the "Detectron" framework [Girshick et al., 2018]. We train an end-to-end Mask R-CNN model with a 64×4d ResNeXt-101-FPN backbone, pre-trained on ImageNet, on our synthetic VEIS dataset. We use mini-batches of size 1 and train the model for 200k iterations, starting with a learning rate of 0.001 and reducing it to 0.0001 after 100k iterations.

5.5.3 Evaluated Methods

In our experiments, we report the results of the following methods:

- **GTA5 [Chen et al., 2017a]:** This baseline denotes a DeepLab model trained on GTA5 by the authors of [Chen et al., 2017a]. We directly report the numbers as provided in [Chen et al., 2017a].
- **GTA5:** This corresponds to our replication of the baseline above. We found our implementation to yield an average accuracy 9.4% higher than the one reported in [Chen et al., 2017a]. As such, this constitutes our true baseline.
- **SYNTHIA:** This refers to a DeepLab model trained on the SYNTHIA [Ros et al., 2016] dataset instead of GTA5.
- **VIPER:** This baseline denotes a DeepLab model trained on the larger VIPER dataset.

- **VEIS:** This corresponds to training a DeepLab model on our new dataset. Note that here we considered all the classes, both foreground and background ones, for semantic segmentation, ignoring the notion of instances.

- **GTA5+VEIS:** This denotes a DeepLab model trained jointly on GTA5 and our new dataset for semantic segmentation.

- **GTA5+VEIS and Pseudo-GT:** For this baseline, we used the results of the GTA5+VEIS baseline to generate pseudo-labels on the real images. We then trained another DeepLab network using these pseudo-labels as ground-truth. In essence, this corresponds to the approach discussed in Section 5.3.2, but without handling the foreground classes in a detection-based manner.

- **Ours:** This corresponds to our method in Section 5.3.1, which relies on the GTA5 synthetic data and makes use of a detection-based model for foreground classes combined with a DeepLab semantic segmentation network for the background ones.

- **Ours and Pseudo-GT:** This consists of using the method above (Ours) to generate pseudo-labels on the real images, and training a DeepLab model from these pseudo-labels, as introduced in Section 5.3.2.

5.5.4 Experimental Results

We now compare the results of the different methods discussed above on the real images of Cityscapes and CamVid. Furthermore, we also compare our approach to the state-of-the-art weakly supervised semantic segmentation and domain adaptation methods on Cityscapes.

In Table 5.2, we provide the results of the methods described above on Cityscapes. The foreground classes are highlighted. In essence, we can see that GTA5 performs better than training DeepLab on the datasets {SYNTHIA,VIPER,VEIS} alone, because these datasets either do not contain all the Cityscapes classes {SYNTHIA,VIPER}, or because they are less realistic {VEIS}. Complementing GTA5 with VEIS {GTA5+VEIS} improves the results by only a small margin, again because of the non-photo-realistic VEIS images. By contrast, using GTA5 and VEIS jointly within our approach (Ours) yields a significant improvement. This is because our detection-based way of dealing with foreground classes is less sensitive to photo-realism, but focuses on shape, which does look natural in our VEIS data. As a matter of fact, our improvement is particularly marked for foreground classes. Finally, while using pseudo-labels from the {GTA5+VEIS} baseline only yields a minor improvement, their use within our framework gives a significant accuracy boost. Some qualitative results are shown in Figure 5.7.

In Table 5.3, we compare our approach with the state-of-the-are weakly-supervised method of [Saleh et al., 2017] and with state-of-the-art domain adaptation methods. The results for these methods were directly taken from their respective papers. Note

Table 5.2: **Comparison of models trained on synthetic data.** All the results are reported on the Cityscapes validation set. Note that (pseudo-GT) indicates the use of unlabeled real images during training. The classes we considered as foreground are denoted by gray rows.

	GTA5 [Chen et al., 2017a]	GTA5	SYNTHIA	VIPER	VEIS	GTA5 + VEIS	GTA5+VEIS pseudo-GT	Ours	Ours pseudo-GT
Road	29.8	80.5	36.7	36.9	70.8	66.2	77.6	71.9	79.8
Sidewalk	16.0	26.0	22.7	19.0	9.5	21.6	26.8	23.8	29.3
Building	56.6	74.7	51.0	74.7	50.9	72.3	75.5	75.5	77.8
Wall	9.2	23.0	0.3	0.0	0.0	15.7	19.4	23.4	24.2
Fence	17.3	9.8	0.1	5.3	0.0	18.3	19.5	14.9	21.6
Pole	13.5	9.1	16.6	7.1	0.3	12.3	4.8	9.3	6.9
Traffic light	13.6	13.4	0.1	10.0	15.6	22.3	18.7	26.7	23.5
Traffic sign	9.8	7.3	9.5	10.1	26.8	23.8	19.8	42.5	44.2
Vegetation	74.9	79.4	72.5	78.7	66.8	78.4	79.5	80.1	80.5
Terrain	6.7	28.6	0.0	13.6	12.7	11.3	21.7	34.0	38.0
Sky	54.3	72.1	78.4	69.6	52.3	74.6	78.9	76.3	76.2
Person	41.9	40.4	47.5	43.0	44.0	48.7	47.3	52.2	52.7
Rider	2.9	5.1	5.6	0.0	14.2	13.3	8.7	28.5	22.2
Car	45.0	77.8	61.4	41.2	60.6	75.1	77.6	76.2	83.0
Truck	3.3	23.0	0.0	20.8	10.2	14.3	23.1	19.6	32.3
Bus	13.1	18.6	13.0	13.9	8.2	21.2	16.1	31.6	41.3
Train	1.3	1.2	0.0	0.0	3.2	2.1	2.2	6.9	27.0
Motorbike	6.0	5.3	3.2	9.1	5.5	24.2	15.6	18.1	19.3
Bicycle	0.0	0.0	3.1	0.0	11.8	7.3	0.0	9.8	27.7
Mean IoU	21.9	31.3	22.1	23.9	24.4	32.8	33.3	38.0	42.5

that, even without seeing the Cityscapes images at all, our approach (Ours) outperforms all these baselines. Using unsupervised Cityscapes images (Ours+pseudo-GT) helps to further improve over the baselines.

The results on CamVid in Table 5.4, where we compare our method to fully-supervised techniques that make use of CamVid images and annotations to train a model, GTA5-based baselines, and the state-of-the-art weakly-supervised method, show a similar trend. Our approach clearly outperforms the weakly-supervised method of [Saleh et al., 2017] and a DeepLab semantic segmentation network trained on synthetic data. In fact, on this dataset, it even outperforms some of the fully supervised methods that rely on annotated CamVid images for training.

5.6 Conclusion

We have introduced an approach to effectively leveraging synthetic training data for semantic segmentation in urban scenes. To this end, we have proposed to handle foreground classes in a detection-based manner, to better account for the fact that existing synthetic datasets represent more accurately the shape of such classes than

Table 5.3: **Comparison to domain adaptation and weakly-supervised methods.** All methods were trained on GTA5, except for [Saleh et al., 2017] which does not use synthetic images. The domain adaptation methods and Ours+Pseudo-GT make use of unlabeled real images during training. The results are reported on the Cityscapes validation set. Note that all the models below use the same backbone architecture as us (DeepLab or FCN8).

Methods	road	side.	buil.	wall	fence	pole	light	sign	Vege.	terr.	sky	person	rider	car	truck	bus	train	motor	bike	mIOU
Fully Sup.	95.8	70.4	85.4	42.7	41.0	21.2	33.7	44.8	86.2	51.4	88.4	58.1	30.1	86.4	43.8	56.7	42.8	33.9	54.8	56.2
Weakly-Sup. [Saleh et al., 2017]	75.9	1.5	41.7	14.1	15.3	6.3	4.4	7.7	58.4	12.6	56.2	16.2	6.1	41.2	22.7	16.6	20.4	15.7	14.9	23.6
[Hoffman et al., 2016]	70.4	32.4	62.1	14.9	5.4	10.9	14.2	2.7	79.2	21.3	64.6	44.1	4.2	70.4	8.0	7.3	0.0	3.5	0.0	27.1
[Zhang et al., 2017]	74.8	22.0	71.7	6.0	11.9	8.4	16.3	11.1	75.7	13.3	66.5	38.0	9.3	55.2	18.8	18.9	0.0	16.8	14.6	28.9
[Chen et al., 2017a]	85.4	31.2	78.6	27.9	22.2	21.9	23.7	11.4	80.7	29.3	68.9	48.5	14.1	78.0	19.1	23.8	9.4	8.3	0.0	35.9
[Hoffman et al., 2017]	85.2	37.2	76.5	21.8	15.0	23.8	22.9	21.5	80.5	31.3	60.7	50.5	9.0	76.9	17.1	28.2	4.5	9.8	0.0	35.4
Ours	71.9	23.8	75.5	23.4	14.9	9.3	26.7	42.5	80.1	34.0	76.3	52.2	28.5	76.2	19.6	31.6	6.9	18.1	9.8	38.0
Ours+Pseudo-GT	79.8	29.3	77.8	, 24.2	21.6	6.9	23.5	44.2	80.5	38.0	76.2	52.7	22.2	83.0	32.3	41.3	27.0	19.3	27.7	42.5

Table 5.4: Comparison of our approach with fully- and weakly-supervised methods on the CamVid test set.

Methods	build.	vege.	sky	car	sign	road	ped.	fence	pole	side.	cyclist	mIOU
SegNet	68.7	52.0	87.0	58.5	13.4	86.2	25.3	17.9	16.0	60.5	24.8	46.4
[Liu and He, 2015]	66.8	66.6	90.1	62.9	21.4	85.8	28.0	17.8	8.3	63.5	8.5	47.2
FCN-8	n/a											52.0
DeepLab-LargeFOV	81.5	74.6	89.0	82.2	42.3	92.2	48.4	27.2	14.3	75.4	50.1	61.6
Dilation8 [Yu and Koltun, 2015b]	82.6	76.2	89.9	84.0	46.9	92.2	56.3	35.8	23.4	75.3	55.5	65.3
Weakly Sup. [Saleh et al., 2017]	58.9	46.4	83.8	26.5	12.0	64.4	8.0	11.3	3.1	1.1	11.0	29.7
GTA5	66.6	53.9	61.4	70.4	32.8	80.9	28.2	24.4	14.6	57.1	0.0	44.6
GTA5+VEIS	73.6	54.2	77.9	66.2	33.6	77.3	26.1	16.0	3.3	48.4	11.9	44.4
Ours	66.3	55.0	61.9	73.4	37.4	82.7	41.4	23.9	9.2	57.7	14.9	47.6
Ours+Pseudo-GT	72.3	55.2	72.6	73.1	37.4	83.9	39.9	33.2	1.2	55.5	12.8	48.8

their texture. Our experiments have demonstrated that our approach outperforms training a standard semantic segmentation network from synthetic data and state-of-the-art domain adaptation techniques. Nevertheless, our approach is orthogonal to domain adaptation. As such, investigating how domain adaptation can be incorporated into our framework could be an interesting avenue for future research.

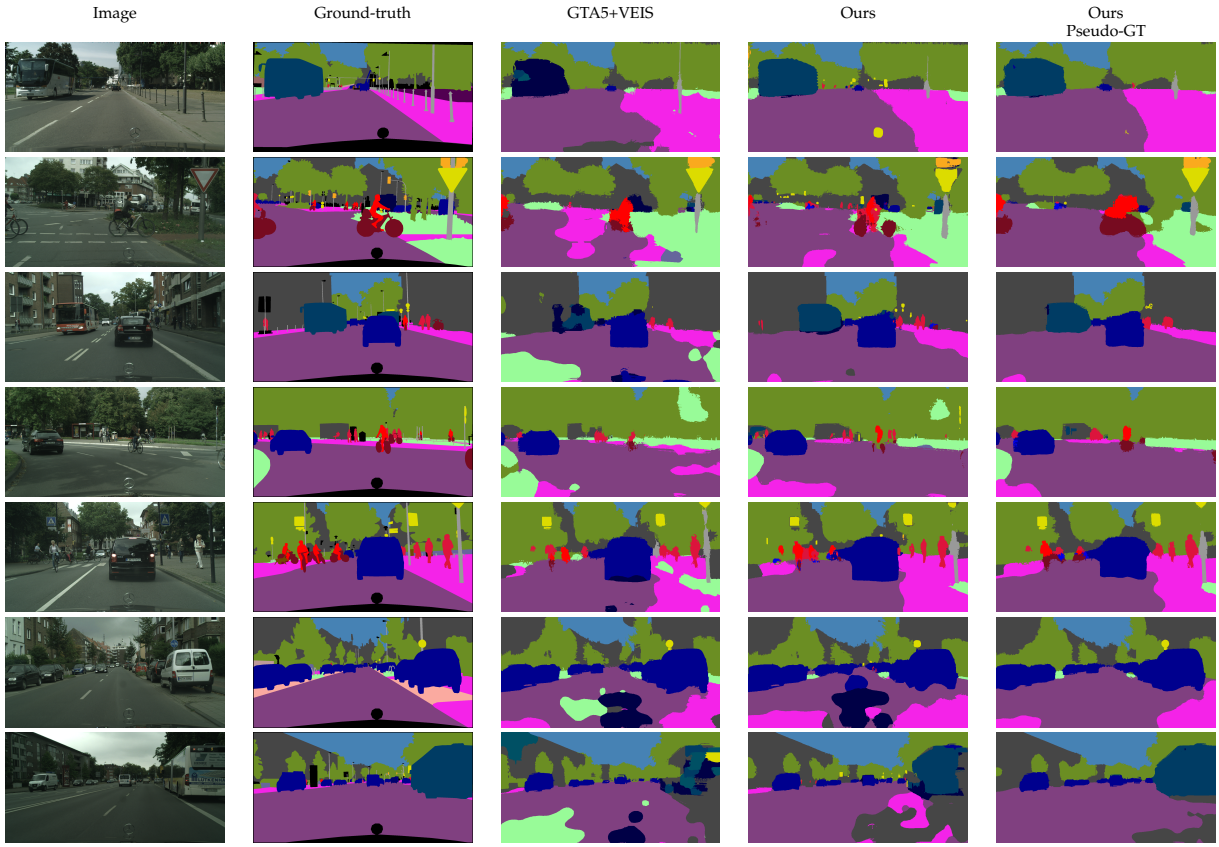


Figure 5.7: Qualitative results on Cityscapes.

Chapter 6

Conclusion

In this thesis, we have tackled the problem of semantic scene segmentation with minimal labeling effort. We have started from one of the weakest level of supervision, image tags, as the only annotation of the training data, and have applied our novel methods on various datasets with their different properties. Although most of the research in this field considers datasets with multiple foreground classes and only one background class in the scene, we have also investigated weakly supervised video semantic segmentation on more challenging datasets such as those with multiple foreground and multiple background classes in the scene. Then, we have investigated using synthetic data where the data is annotated automatically. In this case, we have reached the minimum level of supervision, without requiring to see real images during training.

The major contributions of this thesis are outlined below:

i) We have introduced a novel method to extract accurate foreground/background masks from a pre-trained network, forgoing external objectness modules. The intuition is that a network trained for the task of object recognition extracts features that focus on the objects themselves, and thus has hidden layers with units firing up on foreground objects, but not on background regions. In particular, the proposed method focuses on the fourth and fifth convolution layers of the VGG16 pre-trained network, which provide higher-level information than the first three layers, such as highlighting complete objects or object parts. Then, by making use of a fully-connected Conditional Random Fields (CRF), this information is smoothed out and a binary foreground/background mask is generated which can be incorporated as prior in the network via a weakly-supervised loss function. This work was published in ECCV 2016, Amsterdam.

ii) We have improved our previous contribution to obtain multi-class masks instead of foreground/background ones by introducing a novel method to make use of a pre-trained localization network, which specifically provides information about the location of different object classes in combination with the previous idea of using intermediate convolution layers. The final masks are obtained by making use of a fully-connected Conditional Random Fields (CRF) with higher-order terms, using crisp boundary detection, to smooth the initial pixel-wise probabilities. Then, we incorporated these multi-class masks in a weakly-supervised

loss function to train a Deep Network for the task of semantic segmentation using only image tags as ground-truth annotations. This work was published in TPAMI 2018.

iii) Most of the existing methods, including ours in the previous contributions, are designed to handle multiple foreground classes and a single background class. We have then introduced a novel weakly-supervised video semantic segmentation method that treats all classes, foreground and background ones, equally. To this end, we have proposed a method to rely on class-dependent heatmaps obtained from classifiers trained for image-level recognition, i.e., requiring no pixel-level annotations which provide valuable information about the location of instances/regions of each class. Therefore, we have introduced a weakly-supervised loss function that can exploit them in a two-stream deep architecture which jointly leverages appearance and motion. This work was published in ICCV 2017, Venice.

iv) The use of automatically labeled synthetic data has recently become increasingly popular for semantic scene segmentation. Although these synthetic images are photo-realistic, applying a model trained on these data on a real domain fails because of the domain shift. We have therefore proposed to use synthetic data in a different way to handle this problem. Our approach builds on the observation that foreground and background classes are not affected in the same manner by the domain shift, and thus should be treated differently. Specifically, we use the semantic segmentation network to handle background classes because of their texture realism. By contrast, we utilize the object detection network for the foreground classes due to the fact that their shape looks more natural than their texture in the synthetic domain. Motivated by this fact, we have proposed a simple, yet effective semantic segmentation framework that better leverages synthetic data during training. In essence, the model combines the foreground masks produced by a detector-based instance segmentation network with the pixel-wise predictions of a semantic segmentation network. Furthermore, we have created a virtual environment in the Unity3D framework, called VEIS (Virtual Environment for Instance Segmentation) which automatically annotates synthetic images with instance-level segmentation for foreground classes. It captures urban scenes using a virtual camera mounted on a virtual car. While not highly realistic, when used with a detector-based approach, this data allows us to boost semantic segmentation performance, despite it being of only little use in a standard semantic segmentation framework. This work was published in ECCV 2018, Munich.

6.1 Future Work

In this thesis, we have focused on using weak supervision and synthetic data for semantic scene segmentation. Although there has been a great progress in recent

years using weak annotations and synthetic data for semantic segmentation, there is still a large gap between the performance of the resulting techniques and that working in the fully-supervised setting. This gap becomes even more considerable for complex datasets such as urban scenes. Below, we list some potential future directions based on our research, which expect will help bridge this gap.

i) Incorporating domain adaptation techniques: Our last contribution, i.e., effective use of synthetic data, is orthogonal to domain adaptation. It therefore seems natural to extend our approach to incorporating domain adaptation techniques. Instead of training a network (Mask-RCNN or Deeplab) in the synthetic domain, we can use domain adaptation techniques to first make the distributions of the two domains close to each other and then apply our method of dealing with foreground classes in a detection-based network and dealing with background classes in a semantic segmentation network. Although, in this case, we need to have access to the real domain during training, the approach remains unsupervised and should decrease the gap with fully-supervised techniques.

ii) Generating photo-realistic images: Another potential direction in order to use synthetic data with automatically generated annotation is using style transfer. In fact, one can train a conditional generative adversarial network (GAN) to translate a synthetic image to a photo-realistic one. Then, given the translated image and the corresponding ground-truth one can train a semantic segmentation network in a fully-supervised manner.

iii) Representation learning: In unsupervised learning, it is important to have a model that learns a representation of the data itself which generalizes to data from any domain. For example, when there is no supervision in a driving scenario, it is important to have a model that can learn some common characteristics such as the specific spatial context of the scene. We therefore believe that representation learning can be used as a pre-training step for the final task, i.e., semantic segmentation, or can be used directly during training via an appropriate loss function.

In summary, with the advances in designing weakly-supervised or unsupervised learning methods, in the future, one will be able to utilize large amounts of images and videos, albeit with weak or no annotations to train a more generalized, domain-independent semantic segmentation network. This will make large-scale semantic segmentation much more practical and cost-effective than the current models relying on full supervision, as well as lead to solutions that generalize much better than existing ones, thanks to the use of images depicting a great diversity of scenes.

Bibliography

2018. Notes and assignments for stanford cs class cs231n: Convolutional neural networks for visual recognition. http://cs231n.github.io/convolutional-networks/. (cited on pages xv and 10)

Alexe, B.; Deselaers, T.; and Ferrari, V., 2012. Measuring the objectness of image windows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34, 11 (2012), 2189–2202. (cited on pages xvi, xix, 3, 24, 27, 43, and 44)

Arbeláez, P.; Pont-Tuset, J.; Barron, J.; Marques, F.; and Malik, J., 2014. Multiscale combinatorial grouping. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages xvi, xix, 3, 24, 27, 43, and 44)

Badrinarayanan, V.; Handa, A.; and Cipolla, R., 2015. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, (2015). (cited on page 60)

Batra, D.; Yadollahpour, P.; Guzman-Rivera, A.; and Shakhnarovich, G., 2012. Diverse m-best solutions in markov random fields. In *Computer Vision–ECCV 2012*. Springer. (cited on page 28)

Bearman, A.; Russakovsky, O.; Ferrari, V.; and Fei-Fei, L., 2016. What's the point: Semantic segmentation with point supervision. In *European Conference on Computer Vision (ECCV)*, 549–565. Springer. (cited on pages 2, 3, 22, 23, 24, 27, 32, 33, 35, 36, 37, 38, 40, 41, 43, 53, 57, 58, and 60)

Bertasius, G.; Shi, J.; and Torresani, L., 2014. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. *CoRR*, abs/1412.1123 (2014). http://arxiv.org/abs/1412.1123. (cited on pages 27 and 57)

Brostow, G. J.; Fauqueur, J.; and Cipolla, R., 2009a. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30, 2 (2009), 88–97. (cited on pages 53, 59, 69, 76, and 77)

Brostow, G. J.; Fauqueur, J.; and Cipolla, R., 2009b. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30, 2 (2009), 88–97. (cited on page 60)

Brox, T.; Bruhn, A.; Papenberg, N.; and Weickert, J., 2004. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, 25–36. Springer. (cited on pages 56 and 61)

Brox, T. and Malik, J., 2011. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 33, 3 (2011), 500–513. (cited on page 61)

Carbonneau, M.-A.; Cheplygina, V.; Granger, E.; and Gagnon, G., 2018. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77 (2018), 329–353. (cited on page 17)

Carreira, J. and Sminchisescu, C., 2010. Constrained parametric min-cuts for automatic object segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. (cited on pages 3 and 54)

Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L., 2014. Semantic image segmentation with deep convolutional nets and fully connected crfs. *CoRR*, abs/1412.7062 (2014). http://arxiv.org/abs/1412.7062. (cited on pages 1, 4, 6, 9, 12, 16, 36, 37, 63, 67, 69, 70, 72, and 78)

Chen, Y.; Li, W.; and Van Gool, L., 2017a. Road: Reality oriented adaptation for semantic segmentation of urban scenes. *arXiv preprint arXiv:1711.11556*, (2017). (cited on pages 4, 67, 69, 70, 78, 80, and 81)

Chen, Y.-H.; Chen, W.-Y.; Chen, Y.-T.; Tsai, B.-C.; Wang, Y.-C. F.; and Sun, M., 2017b. No more discrimination: Cross city adaptation of road scene segmenters. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2011–2020. IEEE. (cited on pages 1, 4, 67, and 70)

Cheng, M.-M.; Zhang, Z.; Lin, W.-Y.; and Torr, P., 2014. Bing: Binarized normed gradients for objectness estimation at 300fps. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 3, 24, and 43)

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 1, 53, 55, 59, 60, 69, 76, and 77)

Dai, J.; He, K.; and Sun, J., 2015. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*. (cited on pages 2, 24, and 43)

Dalal, N. and Triggs, B., 2005. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, 886–893. IEEE. (cited on page 60)

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. http://image-net.org. (cited on page 55)

Dosovitskiy, A.; Ros, G.; Codevilla, F.; López, A.; and Koltun, V., 2017. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, (2017). (cited on pages 4, 70, and 75)

Drayer, B. and Brox, T., 2016. Object detection, tracking, and motion segmentation for object-level video segmentation. *arXiv preprint arXiv:1608.03066*, (2016). (cited on pages 3, 54, and 64)

Epic-Games. *Unreal Engine 4.* (cited on page 76)

Everingham, M.; Eslami, S. M. A.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A., 2015. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111, 1 (Jan. 2015), 98–136. (cited on pages 23 and 35)

Farabet, C.; Couprie, C.; Najman, L.; and LeCun, Y., 2013. Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35, 8 (2013), 1915–1929. (cited on pages 1 and 70)

Feichtenhofer, C.; Pinz, A.; and Zisserman, A., 2016. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1933–1941. (cited on pages 55 and 57)

Fragkiadaki, K.; Arbelaez, P.; Felsen, P.; and Malik, J., 2015. Learning to segment moving objects in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4083–4090. (cited on pages 3 and 54)

Ganin, Y. and Lempitsky, V., 2015. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, 1180–1189. (cited on page 70)

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V., 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17, 1 (2016), 2096–2030. (cited on page 70)

Ghodrati, A.; Diba, A.; Pedersoli, M.; Tuytelaars, T.; and Van Gool, L., 2015. Deepproposal: Hunting objects by cascading deep convolutional layers. In *Proceedings of the IEEE International Conference on Computer Vision*, 2578–2586. (cited on page 22)

Girshick, R., 2015. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*. (cited on page 9)

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587. (cited on page 9)

Girshick, R.; Radosavovic, I.; Gkioxari, G.; Dollár, P.; and He, K., 2018. Detectron. https://github.com/facebookresearch/detectron. (cited on page 78)

Glorot, X.; Bordes, A.; and Bengio, Y., 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 315–323. (cited on page 13)

Gould, S.; Rodgers, J.; Cohen, D.; Elidan, G.; and Koller, D., 2008. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80, 3 (2008), 300–316. (cited on page 70)

Hariharan, B.; Arbeláez, P.; Bourdev, L.; Maji, S.; and Malik, J., 2011. Semantic contours from inverse detectors. In *The IEEE International Conference on Computer Vision (ICCV)*. IEEE. (cited on page 36)

Hartmann, G.; Grundmann, M.; Hoffman, J.; Tsai, D.; Kwatra, V.; Madani, O.; Vijayanarasimhan, S.; Essa, I.; Rehg, J.; and Sukthankar, R., 2012. Weakly supervised learning of object segmentations from web-scale video. In *European Conference on Computer Vision*, 198–208. Springer. (cited on pages 3 and 54)

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R., 2017. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2980–2988. IEEE. (cited on pages 6, 69, 73, and 78)

He, K.; Zhang, X.; Ren, S.; and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778. (cited on pages 9 and 15)

Heitz, G. and Koller, D., 2008. Learning spatial context: Using stuff to find things. In *European conference on computer vision*, 30–43. Springer. (cited on page 68)

Hinton, G.; Vinyals, O.; and Dean, J., 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, (2015). (cited on page 70)

Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.-Y.; Isola, P.; Saenko, K.; Efros, A. A.; and Darrell, T., 2017. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, (2017). (cited on pages 4, 67, 69, 70, and 81)

Hoffman, J.; Wang, D.; Yu, F.; and Darrell, T., 2016. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, (2016). (cited on pages 4, 67, 69, 70, and 81)

Hong, S.; Kwak, S.; and Han, B., 2017a. Weakly supervised learning with deep convolutional neural networks for semantic segmentation: Understanding semantic layout of images with minimum human supervision. *IEEE Signal Processing Magazine*, 34, 6 (Nov 2017), 39–49. doi:10.1109/MSP.2017.2742558. (cited on pages xv and 3)

Hong, S.; Yeo, D.; Kwak, S.; Lee, H.; and Han, B., 2017b. Weakly supervised semantic segmentation using web-crawled videos. *arXiv preprint arXiv:1701.00352*, (2017). (cited on page 54)

Huang, Z.; Wang, X.; Wang, J.; Liu, W.; and Wang, J., 2018. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7014–7023. (cited on page 25)

Isola, P.; Zoran, D.; Krishnan, D.; and Adelson, E. H., 2014. Crisp boundary detection using pointwise mutual information. In *European Conference on Computer Vision*, 388–404. Springer. (cited on pages xvi, 5, 23, 32, 33, and 46)

Jain, S. D. and Grauman, K., 2014a. Supervoxel-consistent foreground propagation in video. In *European Conference on Computer Vision*, 656–671. Springer. (cited on pages 1 and 53)

Jain, S. D. and Grauman, K., 2014b. Supervoxel-consistent foreground propagation in video. In *European Conference on Computer Vision*. Springer. (cited on pages 36 and 60)

Jampani, V.; Gadde, R.; and Gehler, P. V., 2017. Video propagation networks. In *Proc. CVPR*, vol. 6, 7. (cited on pages 1, 51, and 53)

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; and Darrell, T., 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, 675–678. ACM. (cited on pages 37 and 61)

Jiang, H.; Wang, J.; Yuan, Z.; Wu, Y.; Zheng, N.; and Li, S., 2013. Salient object detection: A discriminative regional feature integration approach. In *The IEEE conference on computer vision and pattern recognition (CVPR)*. (cited on page 25)

Jin, B.; Segovia, M. V. O.; and Süsstrunk, S., 2017. Webly supervised semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 1705–1714. Ieee. (cited on page 26)

Jin, X.; Li, X.; Xiao, H.; Shen, X.; Lin, Z.; Yang, J.; Chen, Y.; Dong, J.; Liu, L.; Jie, Z.; et al., 2016. Video scene parsing with predictive feature learning. *arXiv preprint arXiv:1612.00119*, (2016). (cited on pages 1, 51, and 53)

Khoreva, A.; Benenson, R.; Hosang, J. H.; Hein, M.; and Schiele, B., 2016. Weakly supervised semantic labelling and instance segmentation. *CoRR*, abs/1603.07485 (2016). http://arxiv.org/abs/1603.07485. (cited on page 2)

Kirillov, A.; He, K.; Girshick, R.; Rother, C.; and Dollár, P., 2018. Panoptic segmentation. *arXiv preprint arXiv:1801.00868*, (2018). (cited on page 73)

Kohli, P.; Torr, P. H.; et al., 2009. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82, 3 (2009), 302–324. (cited on page 45)

Kolesnikov, A. and Lampert, C. H., 2016. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. *CoRR*, abs/1603.06098 (2016). http://arxiv.org/abs/1603.06098. (cited on pages 2, 3, 4, 24, 25, 36, 37, 38, 39, 41, 47, 53, 57, 58, 60, and 61)

Krähenbühl, P. and Koltun, V., 2011. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, 109–117. (cited on pages 18, 19, 32, 35, 58, 59, and 61)

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105. (cited on pages 10 and 15)

Kundu, A.; Vineet, V.; and Koltun, V., 2016. Feature space optimization for semantic video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3168–3175. (cited on pages 1, 51, 53, and 60)

Kuo, W.; Hariharan, B.; and Malik, J., 2015. Deepbox: Learning objectness with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2479–2487. (cited on page 22)

Lafferty, J.; McCallum, A.; and Pereira, F. C., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001). (cited on page 18)

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 11 (1998), 2278–2324. (cited on page 9)

Li, Y.; Shi, J.; and Lin, D., 2018. Low-latency video semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5997–6005. (cited on pages 1, 51, and 53)

Lin, G.; Shen, C.; Van Den Hengel, A.; and Reid, I., 2018. Exploring context with deep structured models for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 40, 6 (2018), 1352–1366. (cited on page 1)

Lin, M.; Chen, Q.; and Yan, S., 2013. Network in network. In *International Conference on Learning Representations (ICLR)*. (cited on page 29)

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L., 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*. Springer. (cited on pages 23 and 36)

Liu, B. and He, X., 2015. Multiclass semantic video segmentation with object-level active inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4286–4294. (cited on pages 53, 62, and 81)

Liu, X.; Tao, D.; Song, M.; Ruan, Y.; Chen, C.; and Bu, J., 2014. Weakly supervised multiclass video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 57–64. (cited on pages 3, 52, and 54)

Long, J.; Shelhamer, E.; and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages xv, 1, 4, 9, 15, 16, 36, 63, 67, 70, and 78)

Mark J. Huiskes, B. T. and Lew, M. S., 2010. New trends and ideas in visual concept detection: The mir flickr retrieval evaluation initiative. In *MIR '10: Proceedings of the 2010 ACM International Conference on Multimedia Information Retrieval* (Philadelphia, USA, 2010), 527–536. ACM, New York, NY, USA. (cited on page 3)

Maron, O. and Lozano-Pérez, T., 1998. A framework for multiple-instance learning. In *Advances in neural information processing systems*, 570–576. (cited on page 17)

Mostajabi, M.; Kolkin, N.; and Shakhnarovich, G., 2016. Diverse sampling for self-supervised learning of semantic segmentation. *arXiv preprint arXiv:1612.01991*, (2016). (cited on page 53)

Mottaghi, R.; Chen, X.; Liu, X.; Cho, N.-G.; Lee, S.-W.; Fidler, S.; Urtasun, R.; and Yuille, A., 2014. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 891–898. (cited on page 70)

Murez, Z.; Kolouri, S.; Kriegman, D.; Ramamoorthi, R.; and Kim, K., 2017. Image to image translation for domain adaptation. *arXiv preprint arXiv:1712.00479*, (2017). (cited on pages 4, 67, and 70)

Noh, H.; Hong, S.; and Han, B., 2015. Learning deconvolution network for semantic segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*. (cited on pages 1, 4, 9, 67, and 70)

Ochs, P.; Malik, J.; and Brox, T., 2014. Segmentation of moving objects by long term video analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36, 6 (2014), 1187–1200. (cited on pages 47 and 63)

Oh, S. J.; Benenson, R.; Khoreva, A.; Akata, Z.; Fritz, M.; Schiele, B.; et al., 2017. Exploiting saliency for object segmentation from image level labels. In *IEEE Conf. Computer Vision and Pattern Recognition*. (cited on pages 26 and 53)

Oquab, M.; Bottou, L.; Laptev, I.; and Sivic, J., 2015a. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 4)

Oquab, M.; Bottou, L.; Laptev, I.; and Sivic, J., 2015b. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 685–694. (cited on page 55)

Overett, G.; Petersson, L.; Brewer, N.; Andersson, L.; and Pettersson, N., 2008. A new pedestrian dataset for supervised learning. In *Intelligent Vehicles Symposium, 2008 IEEE*, 373–378. IEEE. (cited on page 60)

Papandreou, G.; Chen, L.-C.; Murphy, K. P.; and Yuille, A. L., 2015. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*. (cited on pages 1, 2, 3, 24, 35, 36, 37, 38, 39, 41, 42, 53, 57, 60, and 61)

Papazoglou, A. and Ferrari, V., 2013. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision*, 1777–1784. (cited on pages 3, 47, 54, and 63)

Pathak, D.; Krahenbuhl, P.; and Darrell, T., 2015a. Constrained convolutional neural networks for weakly supervised segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*. (cited on pages 2, 3, 23, 24, 36, 37, 38, 39, 40, 41, 53, 57, and 61)

Pathak, D.; Shelhamer, E.; Long, J.; and Darrell, T., 2015b. Fully convolutional multi-class multiple instance learning. In *ICLR Workshop*, 1–4. (cited on pages 2, 17, 24, 32, 33, 36, 37, 53, 57, 58, and 60)

Pinheiro, P. and Collobert, R., 2014. Recurrent convolutional neural networks for scene labeling. In *International conference on machine learning*, 82–90. (cited on page 70)

Pinheiro, P. O. and Collobert, R., 2015. From image-level to pixel-level labeling with convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 2, 3, 22, 24, 25, 27, 32, 33, 34, 36, 37, 38, 39, 40, 41, 43, 53, and 58)

Pourian, N.; Karthikeyan, S.; and Manjunath, B., 2015. Weakly supervised graph based semantic segmentation by learning communities of image-parts. In *The IEEE International Conference on Computer Vision (ICCV)*. (cited on pages 2 and 53)

Prest, A.; Leistner, C.; Civera, J.; Schmid, C.; and Ferrari, V., 2012. Learning object class detectors from weakly annotated video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. (cited on pages 23, 36, 53, and 59)

Qi, X.; Liu, Z.; Shi, J.; Zhao, H.; and Jia, J., 2016. Augmented feedback in semantic segmentation under image level supervision. In *European Conference on Computer Vision*. Springer. (cited on pages 24, 25, 27, 40, 41, 43, and 53)

Qi, X.; Shi, J.; Liu, S.; Liao, R.; and Jia, J., 2015. Semantic segmentation with object clique potential. In *The IEEE International Conference on Computer Vision (ICCV)*. (cited on page 2)

Ren, S.; He, K.; Girshick, R.; and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99. (cited on page 9)

Richter, S. R.; Hayder, Z.; and Koltun, V., 2017. Playing for benchmarks. In *International Conference on Computer Vision (ICCV)*. (cited on pages 4, 6, 69, 70, 74, 75, 76, and 77)

Richter, S. R.; Vineet, V.; Roth, S.; and Koltun, V., 2016. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision*, 102–118. Springer. (cited on pages xvii, 4, 6, 69, 70, 72, 74, 75, and 76)

Robbins, H. and Monro, S., 1985. A stochastic approximation method. In *Herbert Robbins Selected Papers*, 102–109. Springer. (cited on page 15)

Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; and Lopez, A., 2016. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. (cited on pages 4, 6, 69, 70, 74, 76, 77, and 78)

Rumelhart, D. E.; Hinton, G. E.; Williams, R. J.; et al., 1988. Learning representations by back-propagating errors. *Cognitive modeling*, 5, 3 (1988), 1. (cited on pages 10 and 14)

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L., 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115, 3 (2015), 211–252. doi:10.1007/s11263-015-0816-y. (cited on pages 2, 27, 37, 61, and 78)

Saleh, F.; Akbarian, M. S. A.; Salzmann, M.; Petersson, L.; Gould, S.; and Alvarez, J. M., 2016. Built-in foreground/background prior for weakly-supervised semantic segmentation. In *European Conference on Computer Vision*. Springer. (cited on pages 3, 5, 53, 57, 60, and 61)

Saleh, F. S.; Aliakbarian, M. S.; Salzmann, M.; Petersson, L.; and Alvarez, J. M., 2017. Bringing background into the foreground: Making all classes equal in weakly-supervised video semantic segmentation. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2125–2135. IEEE. (cited on pages xx, 3, 6, 70, 78, 79, 80, and 81)

Saleh, F. S.; Aliakbarian, M. S.; Salzmann, M.; Petersson, L.; and Alvarez, J. M., 2018a. Effective use of synthetic data for urban scene semantic segmentation. In *European Conference on Computer Vision*, 86–103. Springer. (cited on page 7)

Saleh, F. S.; Aliakbarian, M. S.; Salzmann, M.; Petersson, L.; Alvarez, J. M.; and Gould, S., 2018b. Incorporating network built-in priors in weakly-supervised semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 40, 6 (2018), 1382–1396. (cited on pages 3, 5, 53, 57, and 61)

Shankar Nagaraja, N.; Schmidt, F. R.; and Brox, T., 2015. Video segmentation with just a few strokes. In *Proceedings of the IEEE International Conference on Computer Vision*, 3235–3243. (cited on pages 1 and 53)

Sharma, A.; Tuzel, O.; and Jacobs, D. W., 2015. Deep hierarchical parsing for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 70)

Shelhamer, E.; Rakelly, K.; Hoffman, J.; and Darrell, T., 2016. Clockwork convnets for video semantic segmentation. In *Computer Vision–ECCV 2016 Workshops*, 852–868. Springer. (cited on pages 1, 51, and 53)

Shimoda, W. and Yanai, K., 2016. Distinct class-specific saliency maps for weakly supervised semantic segmentation. In *European Conference on Computer Vision*. Springer. (cited on pages 24, 25, 38, 39, and 53)

Shotton, J.; Winn, J.; Rother, C.; and Criminisi, A., 2006. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European conference on computer vision*, 1–15. Springer. (cited on page 70)

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, (2014). (cited on pages 9, 15, 22, 27, 36, 37, 55, 57, 61, and 78)

Sun, B. and Saenko, K., 2014. From virtual to reality: Fast adaptation of virtual object detectors to real domains. In *BMVC*, vol. 1, 3. (cited on page 71)

Tang, K.; Sukthankar, R.; Yagnik, J.; and Fei-Fei, L., 2013. Discriminative segment annotation in weakly labeled video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2483–2490. (cited on pages 3, 47, 54, and 63)

Tighe, J. and Lazebnik, S., 2010. Superparsing: scalable nonparametric image parsing with superpixels. In *European conference on computer vision*, 352–365. Springer. (cited on page 70)

Tokmakov, P.; Alahari, K.; and Schmid, C., 2016. Weakly-supervised semantic segmentation using motion cues. In *European Conference on Computer Vision*, 388–404. Springer. (cited on pages 25 and 53)

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M., 2016. Deep end2end voxel2voxel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 17–24. (cited on pages 1, 51, and 53)

Tripathi, S.; Belongie, S.; Hwang, Y.; and Nguyen, T., 2015. Semantic video segmentation: Exploring inference efficiency. In *SoC Design Conference (ISOCC), 2015 International*, 157–158. IEEE. (cited on pages 1, 51, and 53)

Tsai, Y.-H.; Yang, M.-H.; and Black, M. J., 2016a. Video segmentation via object flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3899–3908. (cited on pages 1 and 53)

Tsai, Y.-H.; Zhong, G.; and Yang, M.-H., 2016b. Semantic co-segmentation in videos. In *European Conference on Computer Vision*, 760–775. Springer. (cited on page 54)

Unity3D. *Unity Technologies. Unity Development Platform.* (cited on page 75)

Vezhnevets, A.; Ferrari, V.; and Buhmann, J. M., 2011. Weakly supervised semantic segmentation with a multi-image model. In *The IEEE International Conference on Computer Vision (ICCV)*. IEEE. (cited on pages 2, 24, and 53)

Vezhnevets, A.; Ferrari, V.; and Buhmann, J. M., 2012. Weakly supervised structured output learning for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. (cited on pages 24 and 53)

Vineet, V.; Warrell, J.; and Torr, P. H., 2014. Filter-based mean-field inference for random fields with higher-order terms and product label-spaces. *International Journal of Computer Vision*, 110, 3 (2014), 290–307. (cited on pages 32 and 46)

Wang, H.; Raiko, T.; Lensu, L.; Wang, T.; and Karhunen, J., 2016. Semi-supervised domain adaptation for weakly labeled semantic video object segmentation. *arXiv preprint arXiv:1606.02280*, (2016). (cited on pages 3 and 54)

Wang, P.; Chen, P.; Yuan, Y.; Liu, D.; Huang, Z.; Hou, X.; and Cottrell, G., 2018a. Understanding convolution for semantic segmentation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1451–1460. IEEE. (cited on page 1)

Wang, X.; You, S.; Li, X.; and Ma, H., 2018b. Weakly-supervised semantic segmentation by iteratively mining common object features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1354–1362. (cited on pages 25 and 53)

Wei, Y.; Feng, J.; Liang, X.; Cheng, M.-M.; Zhao, Y.; and Yan, S., 2017. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *IEEE CVPR*, vol. 1, 3. (cited on page 25)

Wei, Y.; Liang, X.; Chen, Y.; Jie, Z.; Xiao, Y.; Zhao, Y.; and Yan, S., 2016a. Learning to segment with image-level annotations. *Pattern Recognition*, (2016). (cited on pages 2, 3, 22, 24, 25, 27, 40, 41, 43, and 53)

Wei, Y.; Liang, X.; Chen, Y.; Shen, X.; Cheng, M.-M.; Feng, J.; Zhao, Y.; and Yan, S., 2016b. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2016). (cited on pages 2, 3, 24, 25, 40, 41, 53, and 61)

Wei, Y.; Xiao, H.; Shi, H.; Jie, Z.; Feng, J.; and Huang, T. S., 2018. 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*, 7268–7277. (cited on page 25)

Xu, J.; Schwing, A.; and Urtasun, R., 2014. Tell me what you see and i will show you where it is. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 2, 24, and 53)

Xu, J.; Schwing, A. G.; and Urtasun, R., 2015. Learning to segment under various forms of weak supervision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 2 and 24)

Yu, F. and Koltun, V., 2015a. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, (2015). (cited on pages xv, 12, 13, and 16)

Yu, F. and Koltun, V., 2015b. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, (2015). (cited on page 81)

Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; and Agrawal, A., 2018. Context encoding for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 1)

Zhang, W.; Zeng, S.; Wang, D.; and Xue, X., 2015a. Weakly supervised semantic segmentation for social images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2718–2726. (cited on page 53)

Zhang, Y.; Chen, X.; Li, J.; Wang, C.; and Xia, C., 2015b. Semantic object segmentation via detection in weakly labeled video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3641–3649. (cited on pages 3, 54, and 64)

Zhang, Y.; David, P.; and Gong, B., 2017. Curriculum domain adaptation for semantic segmentation of urban scenes. In *The IEEE International Conference on Computer Vision (ICCV)*, vol. 2, 6. (cited on pages 4, 67, 69, 70, and 81)

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J., 2017. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2881–2890. (cited on pages 4, 67, 70, and 78)

Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Vineet, V.; Su, Z.; Du, D.; Huang, C.; and Torr, P. H., 2015. Conditional random fields as recurrent neural networks. In *The IEEE International Conference on Computer Vision (ICCV)*. (cited on pages 1 and 70)

Zhong12, G.; Tsai, Y.-H.; and Yang, M.-H. Weakly-supervised video scene co-parsing. (cited on pages 52 and 54)

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A., 2015. Object detectors emerge in deep scene cnns. In *International Conference on Learning Representations (ICLR)*. (cited on page 4)

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A., 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2921–2929. (cited on pages 25, 28, 29, 37, and 55)

Zou, W. and Komodakis, N., 2015. Harf: Hierarchy-associated rich features for salient object detection. In *The IEEE International Conference on Computer Vision (ICCV)*. (cited on page 22)