Context Learning and Weakly Supervised Learning for Semantic Segmentation

Tong Shen

A thesis submitted for the degree of
DOCTOR OF PHILOSOPHY
School of Computer Science
The University of Adelaide

December 2018

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I acknowledge that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Publications

This thesis is based on the content of the following conference papers:

- **Tong Shen**, Guosheng Lin, Chunhua Shen, Ian Reid. Learning Multi-level Region Consistency with Dense Multi-label Networks for Semantic Segmentation. In IJCAI, 2017. (incorporated as Chapter 3)
- **Tong Shen**, Guosheng Lin, Lingqiao Liu, Chunhua Shen, Ian Reid. Weakly Supervised Semantic Segmentation Based on Web Image Co-segmentation. In BMVC, 2017. (incorporated as Chapter 4)
- **Tong Shen**, Guosheng Lin, Chunhua Shen, Ian Reid. Bootstrapping the Performance of Webly Supervised Semantic Segmentation. In CVPR, 2018. (incorporated as Chapter 5)

Acknowledgements

First of all, I would like to express the deepest gratitude to my principle supervisor, Prof. Chunhua Shen for the continuous support of my Ph.D study and related research. He always informed me of new research in the field and provided me with great ideas. When I encountered difficulties, he would always have a insightful discuss with me to solve the issue. It has been a great honour to be his student.

I am also grateful for other members in my advisory team. I would like to thank Prof. Guosheng Lin and Dr. Lingqiao Liu for the in-depth and inspiring discussions. I also want to express my deep gratitude to Prof. Ian Reid for including me in ACRV and supporting my Ph.D research. I feel very lucky to be a member of this brilliant research centre.

I also want to thank my supportive friends and colleagues during my Ph.D study at University of Adelaide. Many thanks to Yuanzhouhan Cao, Vladimir Nekrasov, Ming Cai and Dong Gong for their useful discussions and technical support. I have learnt a lot from them.

Last but not least, I would like to thank my parents who have been very supportive throughout my Ph.D career. I want to give special thanks to my girl friend who encouraged me and helped me when times were tough. I would have never made it this far without their support.

Abstract

This thesis focuses on one of the fundamental problems in computer vision, semantic segmentation, whose task is to predict a semantic label for each pixel of an image. Although semantic segmentation models have been largely improved thanks to the great representative power of deep learning techniques, there are still open questions needed to be discussed. In this thesis, we discuss two problems regarding semantic segmentation, scene consistency and weakly supervised segmentation.

In the first part of the thesis, we discuss the issue of scene consistency in semantic segmentation. This issue comes from the fact that trained models sometimes produce noisy and implausible predictions that are not semantically consistent with the scene or context. By explicitly considering scene consistency both locally and globally, we can narrow down the possible categories for each pixel and generate the desired prediction more easily. In the thesis, we address this issue by introducing a dense multi-label module. In general, multi-label classification refers to the task of assigning multiple labels to a given image. We extend the idea to different levels of the image, and assign multiple labels to different regions of the image. Dense multi-label acts as a constraint to encourage scene consistency locally and globally.

For dense prediction problems such as semantic segmentation, training a model requires densely annotated data as ground-truth, which involves a great amount of human annotation effort and is very time-consuming. Therefore, it is worth investigating semi- or weakly supervised methods that require much less supervision. Particularly, weakly supervised segmentation refers to training the model using only image-level labels, while semi-supervised segmentation refers to using partially annotated data or a small portion of fully annotated data to train. In the thesis, two weakly supervised methods are proposed where only image-level labels are required. The two methods share some similar motivations. First of all, since pixel-level masks are missing in this particular setting, the two methods are all designed to estimate the missing ground-truth and further use them as pseudo ground-truth for training. Secondly, they both use data retrieved from the internet as auxiliary data because web data are cheap to obtain and exist in a large amount. Although there are similarities between these two methods, they are designed from different perspectives. The motivation for the first method is that given a group of images crawled from the internet that belong to the same semantic category, it is a good choice to use co-segmentation to extract the masks of them, which gives us almost free pixel-wise training samples. Those internet images along with the extracted masks are used to train a mask generator to help us estimate the pseudo ground-truth for the training images. The second method is designed as a bi-directional framework between the

target domain and the web domain. The term “bi-directional” refers to the concept that the knowledge learnt from the target domain can be transferred to the web domain and the knowledge encoded in the web domain can be transferred back to the target domain. This kind of interaction between two domains is the core to boost the performance of webly supervised segmentation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In computer vision or robotic vision, how to understand images is a crucial problem. Important tasks such as object detection, image classification, semantic segmentation, are all about how to extract useful features from the raw pixels and utilize the features to give predictions.

Traditional methods use hand-crafted features such as SIFT and HOG (Lowe [1999]; Dalal and Triggs [2005]), which are low-level features used to describe local regions. With the advent of deep neural networks, the features can be learnt from the data in an end-to-end manner. More importantly, the features are learnt hierarchically, which means low-level features are used to describe the local regions whereas high-level features are used to represent more abstract and semantic information. The whole community has greatly benefited from deep neural networks and neural network based methods have achieved a great success in image classification (Simonyan and Zisserman [2015]; He et al. [2016]; Szegedy et al.; Krizhevsky et al. [2012]), object detection (Girshick et al. [2014]; Ren et al. [2015]; Liu et al. [2016]), semantic segmentation (Lin et al. [2017a]; Long et al. [2015]; Chen et al. [2015a]), etc.

In this thesis, we mainly focus on one of the problems mentioned above, semantic segmentation, and address some important issues. Semantic segmentation is crucial when it comes to image understanding. The task is to assign a semantic label to each pixel of an image so that all the classes can be distinguished. Although some methods (Lin et al. [2017a]; Long et al. [2015]; Chen et al. [2015a]) have achieved a great success, there are still open questions out there for us to explore. One of the problems is "scene consistency", which refers to the fact that predictions sometimes are noisy and inconsistent with the context, but we would like to have more plausible predictions. Explicitly encouraging consistency is like imposing a constraint to narrow down possible candidates for each pixel. Another problem discussed in the thesis is how we train networks with partially annotated data, which leads to semi- or weakly supervised problems. In this thesis, we explore these two problems and propose novel solutions.

1.1 Semantic Segmentation

Semantic segmentation is one of the fundamental problems in computer vision whose goal is to classify each pixel of an image into a pre-defined category. For example, an image has a car and a person. Our task is to distinguish the regions of the car, the person and the background. In the early stage of the development of Deep Convolutional Neural Networks (DCNN) based methods (Girshick et al. [2014]; Hariharan et al. [2014]; Carreira et al. [2012]; Cogswell et al. [2014]), two-stage approaches are very common, in which bottom-up image segmentation is usually used to partition the image and a DCNN-based classifier is used to assign categories to the divided regions.

Fully Convolutional Networks (FCN) were proposed by Long et al. [2015]. In this framework, the fully connected layers of a classification network are converted into fully convolutional layers, which makes the network accept various input size and is able to retain spatial information. We are able to perform semantic segmentation in an end-to-end fashion using FCNs. Since then, FCN based methods have dominated the area with different variants (Liu et al. [2015]; Lin et al. [2017a]; Chen et al. [2015a]; Noh et al. [2015]; Zhao et al. [2017]).

Different methods emphasize different aspects of the problem such as enlarging the field of view of the network Chen et al. [2015a], retaining the resolution of the output Noh et al. [2015], and fusing different levels of features Lin et al. [2017a]. The two issues we try to address in the thesis, scene consistency and weakly supervised segmentation, will be explained in the next two sections.

1.2 Scene Consistency

Semantic segmentation is basically a multi-class classification problem but in a dense form, which means that for each pixel, we classify it into the most possible class. In such dense prediction problems, it is very common that predictions are sometimes noisy or inconsistent because the predictions are relatively independent. There are mainly two situations, low-level inconsistency and semantic inconsistency.

In the first situation, it happens when the predictions sharing the similar low level features do not agree with other. The basic assumption behind is that usually, when regions share similar low-level features such as colours and textures, they are likely to belong to the same class. However, if the predictions fail to produce the same result, it could lead to inconsistency.

Another situation is when the predictions are not semantically consistent with the context. For example, in an indoor image, it is likely to have chairs, tables or people, but almost impossible to have planes, trucks or boats. We also call this scene consistency, which means the model should have reasonable predictions given a particular scene.

The first issue can be solved by introducing smoothness to the outputs. Chen et al. [2015a] use Conditional Random Field (CRF) as a post-processing step to smooth the predictions. Similar neighbours tend to have the same label, otherwise there would be great a penalty to the energy. As to the second issue, Liu et al. [2015] propose to include global features to the output to make things globally consistent. Zhao et al. [2017] also propose a pyramid structure to include different levels of features. In this thesis, we will also address this issue in Chapter 3 by introducing a dense multi-label module.

1.3 Weakly Supervised Segmentation

Training a FCN-based network usually requires a dataset with pixel-wise annotation. This indicates that each pixel should be annotated manually, which is very time-consuming. To reduce annotation effort, some methods have been proposed using other types of supervision such as bounding boxes, scribbles and points. Methods relying on pixel-wise labels are usually called fully supervised, whereas other methods relying on other weaker supervisions are called semi- or weakly supervised. To define the terms more formally, we usually call methods that only use image-level labels weakly supervised methods, and call ones that do not use all pixel-level labels semi-supervised methods. In this thesis, we particularly focus on weakly supervised methods requiring only image-level labels.

As pointed in Bearman et al. [2016], it takes averagely 239.7 seconds to process one image to get full supervision, 102.5 seconds to get bounding boxes, and only 20 seconds to get image-level labels. It is obvious that using weak supervisions can greatly save annotation time and make it easier to collect more data. Certainly, the gap between weakly supervised methods and fully supervised methods still exists in terms of performance, but it is worth exploring the situations where only partially annotated data is provided.

Among different types of supervision, image-level labels are the weakest supervision and requires least annotation time. At the same time, it is undoubtedly the most challenging task since there is no evidence indicating the position or extent of objects. The only thing we have access to is the presence of the classes. In this thesis, we focus on this particular task and introduce two weakly supervised methods in Chapter 4 and Chapter 5 respectively.

1.4 Main Contributions

The main contributions of the thesis include a method for scene consistency issue and two frameworks for weakly supervised semantic segmentation. More specific contributions are as follows.

- Semantic image segmentation is a fundamental task in image understanding. Per-pixel semantic labelling of an image benefits greatly from the ability to consider region consistency both locally and globally. However, many Fully Convolutional Network based methods do not impose such consistency, which may give rise to noisy and implausible predictions. We address this issue by proposing a dense multi-label network module that is able to encourage the region consistency at different levels. This simple but effective module can be easily integrated into any semantic segmentation systems. With comprehensive experiments, we show that the dense multi-label can successfully remove the implausible labels and clear the confusion so as to boost the performance of semantic segmentation systems.

- We propose a novel method for weakly supervised semantic segmentation with only image-level labels. The method relies on a large scale co-segmentation framework that can produce object masks for a group of images containing objects belonging to the same semantic class. We first retrieve images from search engines, e.g. Flickr and Google, using semantic class names as queries, e.g. class names in PASCAL VOC 2012. We then use high quality masks produced by co-segmentation on the retrieved images as well as the target dataset images with image level labels to train segmentation networks. We obtain IoU 56.9 on test set of PASCAL VOC 2012, which reaches state of the art performance.

- We develop a bi-directional framework for training a high-quality pixel-level classifier for semantic segmentation, using only image-level class labels as the provided ground-truth. Our method is formulated as a two-stage approach in which we first aim to create accurate pixel-level masks for the training images via a bootstrapping process, and then use these now-accurately segmented images as a proxy ground-truth in a more standard supervised setting. The key driver for our work is that in the target dataset we typically have reliable ground-truth image-level labels, while data crawled from the web may have unreliable labels, but can be filtered to comprise only easy images to segment, therefore having reliable boundaries. These two forms of information are complementary and we use this observation to build a novel bi-directional transfer learning framework. This framework transfers knowledge between two domains, target domain and web domain, bootstrapping the performance of weakly supervised semantic segmentation. Conducting experiments on the popular benchmark dataset PASCAL VOC 2012 based on both a VGG16 network and on ResNet50, we reach state-of-the-art performance with scores of 60.2% IoU and 63.9% IoU respectively

1.5 Thesis Overview

The structure of the thesis is organized as follows.

Chapter 2 presents a literature review on DCNNs including basic building blocks and some popular architectures. There is also related work introduced about semantic segmentation and related problems.

In Chapter 3, we will address the issue of scene consistency by introducing a dense multi-label module, which enforces scene consistency locally and globally.

In Chapter 4, we will explore the problem of weakly supervised segmentation. We introduce a framework that utilizes co-segmentation to estimate pixel-wise masks that are not available in weakly supervised settings.

In Chapter 5, we will introduce a bi-directional framework for weakly supervised segmentation. The framework transfers knowledge between two domains, the target domain and the web domain, and boosts the performance.

In Chapter 6, we will give a brief conclusion to the thesis and suggest some future work for the two main problems discussed in the thesis.

Chapter 2

Literature Review

In this chapter, we first review Convolutional Neural Networks (CNNs) including some common components and architectures. Then we review semantic segmentation and introduce some related work.

2.1 Convolutional Neural Networks

2.1.1 Introduction

Artificial Neural Networks (ANN) were originally inspired by biological neural networks and have been studied for decades. An ANN can be viewed as a highly non-linear function that maps some inputs to some outputs. ANNs usually consist of hidden nodes, weights and activation functions. Hidden nodes can form into hidden layers and hidden layers can interact with each other. When training an ANN, we normally provide training data with inputs and corresponding outputs and alter the weights to minimize the loss and fit into the training data.

In the early stage of ANNs, networks are simple and shallow, which means only a few hidden layers are involved. For example, Perceptron is a very simple algorithm that deals with binary classification problems. In this model, the output is basically a linear combination of the inputs without any hidden layers. Multilayer Perceptron (MLP) is a network with one or more hidden layers. These networks have limited capacity and representation power. Therefore, it is difficult to apply them to complex problems. Besides, since nodes are fully connected between layers, it is difficult to scale up to image-related problems due to the huge number of parameters involved.

LeCun et al. [1998] first proposed a CNN, called LetNet, to solve the image classification problem. Different from conventional neural networks with all fully connected layers, a CNN consists of convolutional layers that partially connected. In this way, weights can be shared across the image and the parameters are largely reduced. In addition, sub-sampling operations were also introduced to reduce the size of feature maps and increase receptive field. The main idea of CNNs has influenced the community greatly. Krizhevsky et al. [2012] first proposed a CNN, called AlexNet, for large scale image classification, which showed outstanding performance and indicated great potential to computer vision problems. With more attention drawn for CNNs, more good structures, GoogLeNet Szegedy et al. [2015], VGGNet Simonyan and Zisserman [2015], ResNet He et al. [2016], etc., have been proposed in the recently several years. More techniques, skip connection He et al. [2016], batch normalization Ioffe and Szegedy [2015], etc., have also been introduced to accelerate the optimization process and increase the depth.

The newly proposed networks such as AlexNet, VGGNet, ResNet, etc. have much more layers than the conventional neural networks. Variants of ResNet have 101 layers, 152 layers or even 1000 layers. The community stepped into the deep learning era and many methods have been proposed for various vision problems. In the next two sub-sections, we review some important components of CNNs and some popular architectures.

2.1.2 Components

A CNN is usually comprised of several convolutional layers with some activation layers, pooling layers and fully connected layers. There are also auxiliary layers such as batch normalization layers and dropout layers, to accelerate convergence or increase generalization capacity. In this section, we give a brief review of these basic building blocks.

- **Convolutional Layer** is the most basic building block of CNNs. A typical convolutional layer has dimensions of $C_{out} \times C_{in} \times K \times K$ for 2D CNNs, where C_{in} indicates the number of channels of the previous layer; C_{out} indicates the number of filters and K is the size of the kernel. K is usually 1 or 3. Although large kernels, 5 or 7, are also used, it is more common to replace a large kernel with a series 3×3 kernels to reduce computation. Compared with conventional neural networks where nodes are fully connected, convolutional layers only make the feature maps partially connected. Taking a 3×3 kernel as an example, a point of the output only corresponds to 9 points of the input instead of the whole input. In addition, the weights are shared across all the nodes. In this way, both the computation and the number of parameters are largely decreased. To further decrease the computation, convolutional layers can also be used as a sub-sampler to reduce the spatial dimensions of the feature maps. To achieve this, we simply use a larger stride, larger than 1, to skip some convolution operations. A stride of 2 will result in half size of the input size.

- **Pooling Layer** is another way of sub-sampling feature maps. Similarly, a pooling layer uses different strides to reduce the resolution of the feature maps. There are two types of pooling layers commonly used, average pooling and max pooling, which refer to average and max operation on a local region respectively. The role of pooling layers is to reduce the spatial dimensions of the feature maps so as to reduce the computation, and also to introduce translation invariance, which makes the network insensitive to small shift.

- **Fully Connected Layer** makes a connection between every input point and output point. The layer is usually used before the classifier of a CNN to transform the spatial information to global information for classification. It is widely used in popular structures such as AlexNet Krizhevsky et al. [2012] and VGGNet Simonyan and Zisserman [2015]. However, the downside of this layer is that it might bring a big computational burden.

- **Activation Layer** is usually a non-linear function that allows networks to deal with more complex problems. Sigmoid activation layer used to be popular in the early methods, but it turned out that it gets saturated easily and makes optimization hard. Rectified Linear Unit (ReLU) was proposed by Nair and Hinton [2010] to deal with the issue. Now, ReLU has become a standard activation layer following convolutional layers or fully connected layers. ReLU makes optimization easier and deeper networks possible. There are also variants of ReLU such as Leaky ReLU Maas et al. [2013] and ELU Clevert et al. [2016].

- **Dropout Layer** provides a way to prevent overfitting Srivastava et al. [2014]. During training, it randomly sets some nodes to zero, which, in other words, prevents them from activating. This random drop-out process is able to prevent overfitting in a sense that each time only a few number of nodes are activated, which introduces sparsity to the model and reduces the complexity. During inference, the behaviour is a bit different. There is no node prevented from activating, but the outputs should be multiplied by a scalar to keep the magnitude.

- **Batch Normalization Layer** is an effective layer for accelerating training speed and preventing overfitting. In the original paper Ioffe and Szegedy [2015], the authors discuss an issue called "internal covariate shift", which is the main cause of the slow training. They address the issue by proposing batch normalization layer to normalize the input of a layer within a batch. By using batch normalization, the training time is significantly decreased and the network is robust to different initializations. Additionally, they also claim that the layer can act as a regularizer making it an alternative to dropout.

2.1.3 Architectures

Since AlexNet Krizhevsky et al. [2012] won the "ImageNet Large Scale Visual Recognition Challenge" (ILSVRC) Russakovsky et al. [2015] in 2012 by a big margin, there have been several well-designed network architectures proposed. In this section, we review some popular architectures that have been widely used in the computer vision community.

- **AlexNet** was proposed by Krizhevsky et al. [2012] and was the winner of ILSVRC in 2012. The network has five convolutional layers and three fully connected layers. There are also max pooling layers in between to down-sample the features. The fully-connected layer before the classifier has 4096 neurons.

- **VGGNet** was proposed by Simonyan and Zisserman [2015]. There are two versions, VGG16 and VGG19, which contain 16 and 19 convolutional layers respectively. There are five groups in VGGNet. each group has several 3x3 convolutional layers followed by a ReLU. The features are downsampled after each group, and correspondingly the number of channels is doubled in the next group. There are also 4096 neurons before the classifier. VGGNet is much deeper than AlexNet and achieves better performance in various vision tasks.

- **GoogLeNet** Szegedy et al. [2015] was the winner of ILSVRC in 2014. The network is built with building blocks called "Inception modules". In each Inception module, there are parallel kernels in different sizes and there are also 1x1 kernels for dimension reduction.

- **ResNet** He et al. [2016] won ILSVRC in 2015. There are different variants based on the depth of the networks, ResNet50, ResNet101, and ResNet152. Obviously ResNet is much deeper than all the previous architectures. The issue for the other relatively shallow architectures was that with the depth of the network increasing, it is difficult to back propagate gradients to the early layers. In ResNet, a residual module was proposed as the basic building unit. In the module, there is a skip connection from the input directly to the output, which makes the gradient pass backward very easily. Besides, a bottle neck design is adopted to reduce the dimension of the input and increase the dimension of the output, which largely reduces the number of parameters and improves the speed. Batch Normalization is also an important component of the residual block that helps to accelerate the training.

2.2 Semantic Segmentation

2.2.1 Introduction

Semantic segmentation is one of the fundamental problems in computer vision that aims to predict a semantic label for all the pixels given an image. Semantic segmentation gives us good understanding of the scene and other computer vision tasks can also benefit from the result.

In the early stage, semantic segmentation methods rely on hand-crafted features and separate classifiers (Shotton et al. [2008]; Carreira et al. [2012]; Krähenbühl and Koltun [2011]; Shotton et al. [2006]).

Later on, with the success of deep neural networks, hand-crafted features are outperformed by the deep features learnt from large image datasets. Some methods (Farabet et al. [2013]; Girshick et al. [2014]; Hariharan et al. [2014]) are proposed to use deep networks as either a feature extractor or a deep classifier to improve semantic segmentation performance. However, those methods still treat deep networks as a separate module for a specific role. Long et al. [2015] first propose an end-to-end framework for semantic segmentation called Fully Convolutional Networks (FCNs). The idea is to convert the fully connected layers in classification networks to convolutional layers so that the output will contain spatial information. Since then, FCN-based frameworks have achieved remarkable success in semantic segmentation. In the following sections, we will only focus on FCN-based methods.

2.2.2 Fully Supervised Methods

In Long et al. [2015], the classification network is firstly modified by replacing the fully-connected layers to convolutional layers. The last classifier layer is also replaced by a newly initialized layer to fit in the new segmentation task. The model is then finetuned on the target dataset. In order to train such a model, pixel-level labels are usually required, which means all the pixels should be annotated for all the training images. The training scheme is called fully supervised training because the images are densely annotated to include all the information.

Since the original FCN model, there have been variants of FCN model that have dominated the field of semantic segmentation. Noh et al. [2015] propose an encoder-decoder framework using VGG16 as the backbone. First, the image goes into an encoder converted from a standard VGG16 and the features are downsampled 5 times. Then a mirrored structure is used as the decoder to upsample the features. Instead of further downsampling the features, the decoder uses unpooling layers or deconvolutional layers to upsample the features and reconstruct the details. The final prediction is given by the output of the decoder.

Chen et al. [2016] introduce a FCN-based model with atrous convolution and fully connected CRFs. Atrous convolution is used to enlarge the field-of-view of the kernels without increasing the number of parameters. It allows use to explicitly control the resolution of network. Fully connected CRF is used to post-process the prediction so that the final prediction is smooth and has better boundaries.

Lin et al. [2017b] address the problem of explicitly modelling context information in semantic segmentation. In order to make predictions semantically more consistent, they define several pairwise potential functions to model the relationships between nodes. Unlike Chen et al. [2016] where CRF is used as a post-processing step, this framework combines the FCN and CRF together as a unified network to formulate patch-patch context between image regions.

Lin et al. [2017a] propose an end-to-end network to explicitly reuse low-level features to help reconstruct details for high-level features at low resolution. The paper introduces a chained residual pooling and a residual convolution unit to process and fuse the features from multiple paths.

2.2.3 Semi- and Weakly Supervised Methods

In the last section, we have discussed some popular fully supervised methods. However, a problem we are facing for fully supervised methods is that pixel-level labels are always required, which is very expensive in term of annotation effort. To this end, some weakly and semi supervised methods have been proposed to alleviate this issue.

There are various types of supervision that have been explored such as points, bounding boxes, scribbles. Bearman et al. [2016] use point supervision to replace pixel-level masks. Objects are annotated with points and a specific loss is designed for this particular supervision. In addition, objectness priors are also integrated to impose the smoothness and consistency. Dai et al. [2015] train segmentation networks with only bounding boxes. The motivation is that bounding boxes accurately indicate the location as well as the extent of objects. With the help of some mask proposal method (e.g. Multiscale Combinatorial Grouping), it is possible to find pixel-level masks for the objects. Lin et al. [2016a] use scribbles to supervise the network. Scribbles are not as accurate as bounding boxes, but it is easier to annotate images with scribbles. In this work, they use scribble-based and network-based unary terms and pairwise term to formulate a graphical model and find the pixel-level mask.

Among different types of supervision, the weakest supervision is image-level labels. There is no information indicating the location or size of the objects. The only information is the presence of the objects. Pinheiro and Collobert [2015] treat weakly supervised segmentation as a multiple instance learning problem where every training image is known to have at least one pixel belonging to the class shown in the image-level label. Then they use the attention learnt by image classification as well as smoothing priors to generate pixel-level predictions. Pathak et al. [2015] formulate the weakly supervised segmentation as a constrained optimization problem and propose some constraints such as suppression constraint, foreground and background constraint, and size constraint. Kolesnikov and Lampert [2016] present an end-to-end network for weakly supervised segmentation where three losses, seed, expand and constrain, are proposed to supervise the model. Seed loss supervises the model with network attentions; expand loss controls how to aggregate the heat maps; constraint loss makes the prediction respect the boundaries of the objects.

There are some special weakly supervised methods that use extra web data as supervision. Since the web data are almost unlimited and easy to obtain, using web data does not involve much human effort. The methods that use web data are sometimes called webly supervised methods. Wei et al. [2017b] propose a simple-to-complex framework for weakly supervised segmentation. They start by retrieving images from the internet and use saliency to extract the foreground. This retrieved dataset is then used to train a FCN. Then the model is used to generate masks for the target dataset and further enhanced by using image-level labels. Jin et al. [2017] is another similar framework where some simple images are first retrieved from the internet and used to learn the features of all classes. Then the model is used to estimate masks for the target dataset. Hong et al. [2017] present a weakly supervised method using web videos. In this work, a large number of videos are crawled from the internet. The frames extracted from the videos are further filtered to remove noise. Then they formulate an optimization problem to extract the foreground of the video frames using network attentions, motion cues and smoothness terms.

2.2.4 Scene Consistency Issue

Although many FCN-based models handle the problem of semantic segmentation well, there still exists an issue of scene consistency. Scene consistency, or region consistency, refers to consistency of predictions within a region. A model that fails to retain scene consistency would predict noisy results.

There have been some papers trying to address the issue. Lin et al. [2017b] exploit context information by introducing pairwise potentials into the framework. There is no explicit consistency imposed, but the final predictions would benefit from considering the neighbours. Liu et al. [2015] address the problem by concatenating global features to the dense features. In such way, the predictions will not only rely on the local features but also the global features. Zhao et al. [2017] introduce a pyramid pooling module to consider different levels of features. The different levels of features are extracted using different sliding windows and concatenated together along with the dense features. The final predictions become more consistent with the scene. Wang et al. [2016] propose a learnable histogram layer that learn to represent the scene by a histogram. The histogram can then be used as global context constraint to impose on the final predictions.

Chapter 3

Learning Multi-level Region Consistency with Dense Multi-label Networks

3.1 Introduction

In this chapter, we will discuss the problem of scene consistency. We have seen the remarkable performance achieved by FCN-based models (Dai et al. [2015]; Chen et al. [2015a]; Lin et al. [2016b]; Chen et al. [2016]). These models are simple and effective because of the powerful capacity of CNNs and being able to be trained end-to-end. However, most existing methods do not have the mechanism to enforce the region consistency, which plays an important role in semantic segmentation. Consider, for example, Figure 3.1, in which the lower left image is the output of a vanilla FCN, whose prediction contains some noisy labels that do not appear in the ground truth. With enforced region consistency, we can simply eliminate those implausible labels and clear the confusion. Our aim in this chapter is to introduce constraints to encourage this consistency.

Our proposal is both simple and effective: we argue that the region consistency in a certain region can be formulated as a multi-label classification problem. Multi-label classification has also been widely studied (Jiang [2016]; Wei et al. [2016]; Guo and Gu [2011]), whose task is to assign one or more labels to the image. By performing

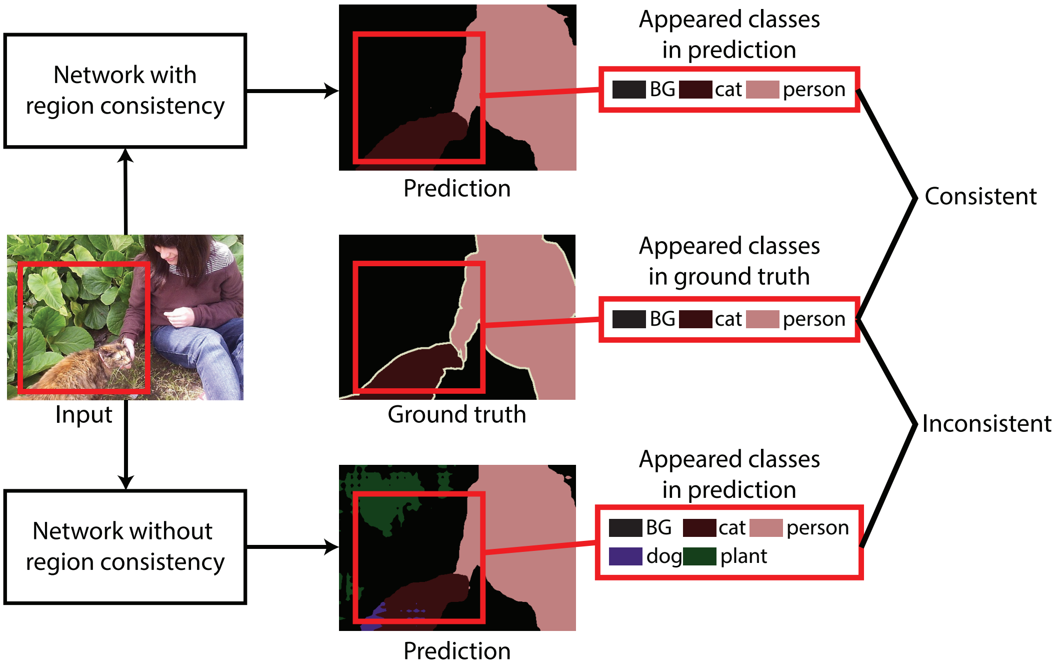

Network with region consistency
Prediction
Appeared classes in prediction
BG cat person
Consistent
Input
Ground truth
Appeared classes in ground truth
BG cat person
Inconsistent
Network without region consistency
Prediction
Appeared classes in prediction
BG cat person dog plant

Figure 3.1: Illustration of region consistency. For a region in the input image, which is coloured in red, the corresponding part in the ground truth contains only three classes. In the network without region consistency, there are five classes that appear. If we explicitly encourage the consistency, those unlikely classes will be eliminated and the prediction will be better as shown on top.

multi-label classification in a region, we can allow the model to suggest which labels are likely within the broad context of the region, and use this information to suppress implausible classes predicted without reference to the broader context, thereby improving scene consistency. While typical multi-label problems are formulated as whole-image inference, we adapt this approach to dense prediction problems such as semantic segmentation, by introducing dense multi-label prediction for image regions of various sizes.

Dense multi-label prediction is performed in a sliding window fashion: the classification for each spatial point is influenced by the network prediction and by the multi-label result for the surrounding window. By employing different window sizes, we are able to construct a multi-level structure for dense multi-label and enforce the

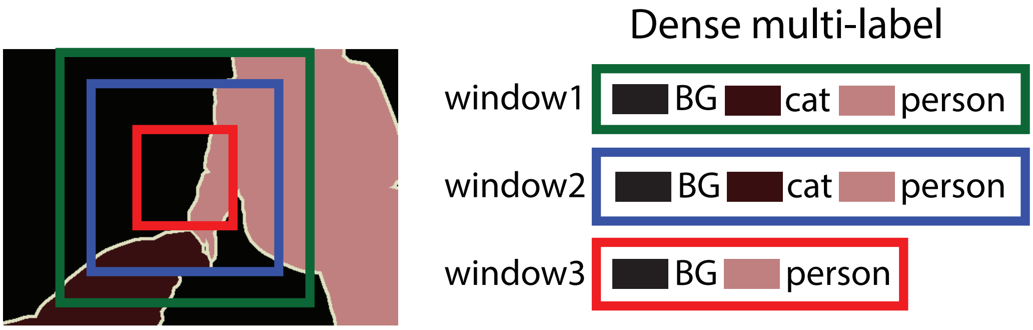


Figure 3.2: Illustration of dense multi-label with multi-level. Windows in different colours indicate different regions for dense multi-label classification.

region consistency at different levels both locally and globally. Figure 3.2 is an illustration of dense multi-label at multiple windows sizes.Here we use three windows of different sizes. The red window, the smallest, focuses more on the local region consistency, while the green window, the largest, is responsible for global region consistency. The other one, in blue, is for mid-level consistency. By sliding the windows to consider each spatial point, we perform multi-label densely at different level, encouraging the segmentation predictor to give predictions that are consistent with the dense multi-label prediction.

Main contributions can be summarised as follows:

- We address the problem of region consistency in semantic segmentation by proposing a dense multi-label module to achieve the goal of retaining region consistency, which is simple and effective. We also introduce a multi-level structure for dense multi-label to preserve region consistency both locally and globally.

- We evaluate our method on four popular semantic segmentation datasets including NYUDv2, SUN-RGBD, PASCAL-Context and ADE 20k, and achieve promising results. We also give analysis on how dense multi-label can remove the implausible labels, clear confusion and effectively boost the segmentation systems.

3.2 Background

Semantic segmentation has been widely studied (Girshick et al. [2014]; Carreira et al. [2012]; Hariharan et al. [2014]; Yadollahpour et al. [2013]; Farabet et al. [2013]; Cogswell et al. [2014]). Early CNN based methods rely on region proposals or superpixels. They make segmentation prediction by classifying these local features.

More recently, with Long et al. [2015] introducing applying FCNs to semantic segmentation, the FCN-based segmentation models (Dai et al. [2015]; Hong et al. [2015]; Shen and Zeng [2016]; Chen et al. [2015a]; Lin et al. [2016b]; Chen et al. [2016]) have become popular. Long et al. [2015] convert the last fully connected layers into convolutional layers thus make the CNN accept abitrary input size. Since the output retains the spatial information, it is straightforward to train the network jointly in an end-to-end fashion. They also introduce skip architecture to combine features from different levels. Chen et al. [2015a] modify the original FCN by introducing dilated kernels, in which kernels are inserted with zeros, to enable large field of view and Fully Connected CRF to refine outputs. Lin et al. [2016b] introduce a joint training model with CRFs. In this work, CRFs are not simply used for smoothness as in Chen et al. [2015a], but a more general term to learn context information to help boost the unary performance. Liu et al. [2015] utilise global features to improve semantic segmentation. They extract global features from different levels and fuse them by using L2 normalization layer. Our method is different from those. We attempt to improve the performance of segmentation by enforcing region consistency using dense multi-label.

Multi-label classification has also been widely studied. Traditional methods are based on graphical models (Xue et al. [2011]; Guo and Gu [2011]), while the recent studies benefit more from CNNs Wei et al. [2014]; Jiang [2016]; Gong et al. [2013]. Gong et al. [2013] transform a single-label classification model into multi-label classification model and use ranking loss to train the model. Wei et al. [2014] also use the transfer learning from single-label classification models. They perform the multi-label classification by first generating the object hypotheses and the fusing predictions as the final prediction for the whole image. Jiang [2016] propose a unified framework for multi-label classification by using CNN and Recurrent Neural Network (RNN).

Here we propose a dense multi-label module to take advantage of multi-label classification and integrate it into semantic segmentation systems. Dense multi-label is performed in a sliding window fashion and treats all area in a window as multi-label classification. Experiments show that dense multi-label can help to keep the scene consistency, clear confusion and boost the performance of semantic segmentation.

3.3 Methods

3.3.1 Dense Multi-label

Multi-label classification is a task where each image can have more than one label, unlike a multi-class classification problem (Simonyan and Zisserman [2015]; Szegedy et al.; He et al. [2016]) whose goal is to assign only one label to the image. This is more natural in reality because for majority of images, objects are not isolated, instead they are in context with other objects or the scene. Multi-label classification gives us more information of the image.

For a dense prediction task such as segmentation, it treats every spatial point as a multi-class classification problem, where the point is assigned with one of the categories. As shown in the upper part of Figure 3.3, the model predicts scores for each class and picks the highest one. The ground truth is an one-hot vector correspondingly. For a dense multi-label problem, each spatial point will be assigned with several labels to show what labels appear in the a certain window centered at

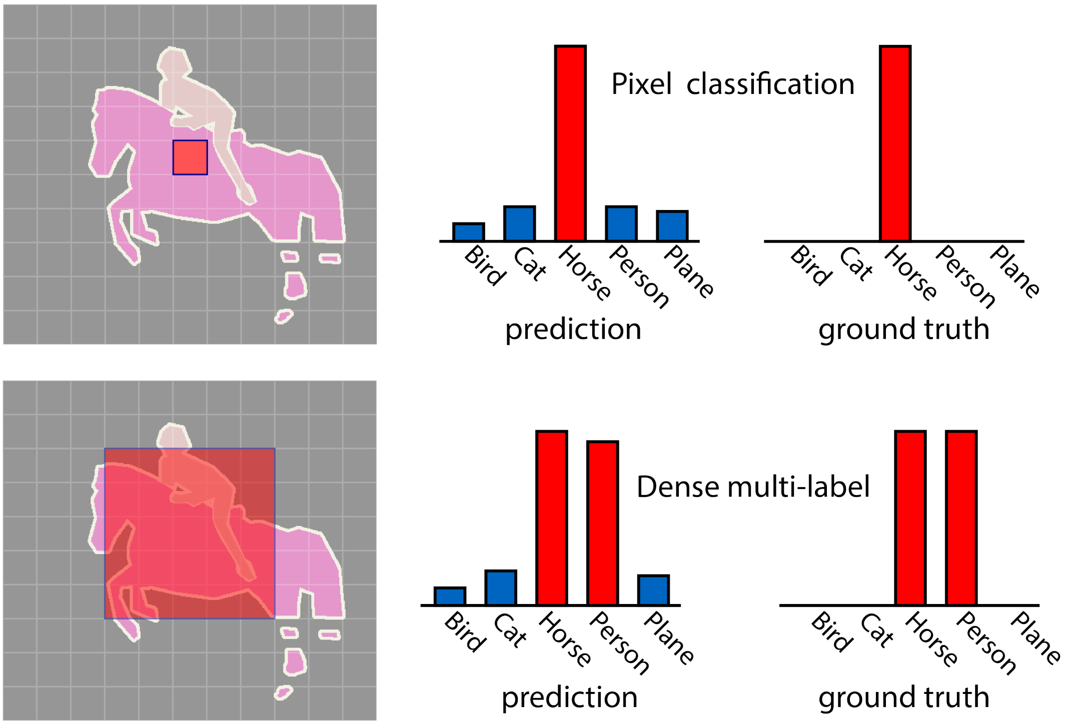


Figure 3.3: An illustration of differences between pixel classification and dense multi-label prediction. In pixel classification, we treat each spatial point as a single-label classification problem where only one class is supposed to get very high confidence; dense multi-label focuses on label concurrence where the labels that appear in the region will have equally high confidence.

this point. As shown in lower part of Figure 3.3, there are two classes being predicted with high confidence and the ground truth is given by a “multiple hot” vector.

Here we propose a method to learn a dense multi-label system and a segmentation system at the same time. We aim at using dense multi-label to suppress the implausible classes and encourage appropriate classes so as to retain the region consistency for the segmentation prediction both globally and locally. In the next section, more details of the whole framework will be provided.

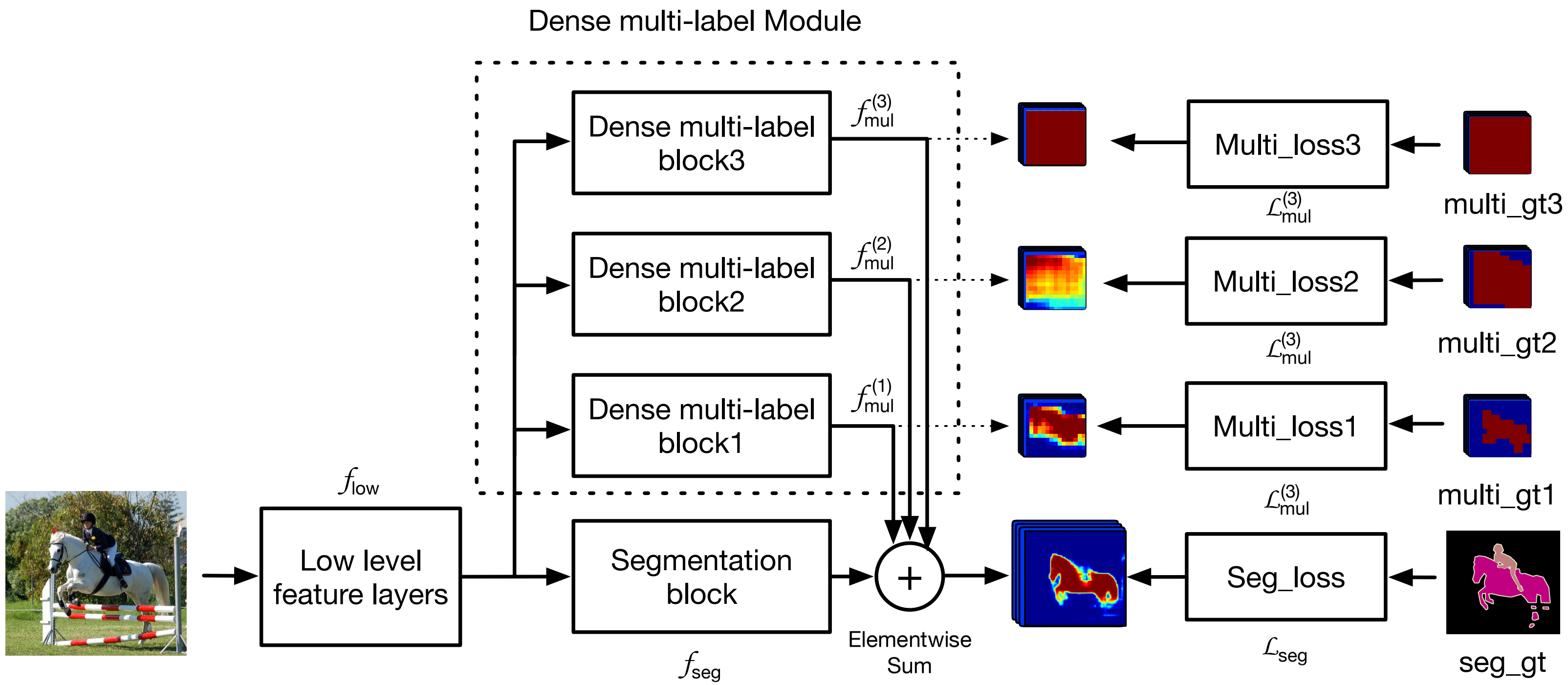


Figure 3.4: Illustration of the framework with dense multi-label module. The input image is first passed into low level feature layers, which are shared by the following blocks. Then the feature maps are fed into the segmentation block and three dense multi-label blocks. The element-wise sum will sum up the features from the blocks and make the final prediction. Apart from the segmentation loss, each dense multi-label block also has its own multi-label loss to guide the training.

3.3.2 Overview of Framework

An overview of the structure is shown in Figure 3.4, with the part in the dashed-line rectangle being the dense multi-label module. Without it, the network simply becomes a FCN. The input image is first fed into several low level feature layers that are shared by the following blocks. Then apart from going into the segmentation block, the features also enter three blocks for dense multi-label prediction. The outputs of theses blocks are merged element-wise for the final prediction.

In the training phase, the network is guided by four loss functions: the segmentation loss and three dense multi-label losses. We use softmax loss for the segmentation path, and use logistic loss for all the dense multi-label blocks.

The dense multi-label blocks have different window sizes for performing dense multi-label prediction within different contexts. With this multi-level structure, we are able to retain region consistency both locally and globally.

Let $\boldsymbol{x}$ denote the image. The process of the low level feature block can be described as:

$$\boldsymbol{o} = f_{low}(\boldsymbol{x}; \boldsymbol{\theta}_{low}), \tag{3.1}$$

where $\boldsymbol{o}$ is the output and $\boldsymbol{\theta}_{low}$ the layer parameters.

The dense multi-label blocks and the segmentation block are defined as:

$$\boldsymbol{m}^{(j)} = f^{(j)}_{mul}(\boldsymbol{o}; \boldsymbol{\theta}^{(j)}_{mul}), j \in \{1, 2, 3\} \tag{3.2}$$

$$\boldsymbol{s} = f_{seg}(\boldsymbol{o}; \boldsymbol{\theta}_{seg}), \tag{3.3}$$

where $\boldsymbol{m}^{(j)}$ and $\boldsymbol{s}$ denote the output of jth multi-label block and the output of segmentation respectively. $\boldsymbol{\theta}^{(j)}_{mul}$ and $\boldsymbol{\theta}_{seg}$ are layer parameters.

The final prediction is:

$$\boldsymbol{p} = \boldsymbol{s} + \boldsymbol{m}^{(1)} + \boldsymbol{m}^{(2)} + \boldsymbol{m}^{(3)}, \tag{3.4}$$

where $\boldsymbol{p}$ is the fused score for segmentation.

For the loss functions, we use logistic loss for the prediction of dense multi label blocks, $\boldsymbol{m}^{(1)}$,$\boldsymbol{m}^{(2)}$ and $\boldsymbol{m}^{(3)}$; softmax loss is used for final prediction $\boldsymbol{p}$. Let m_{ik} be the out of a dense multi-label block at ith position for kth class, and y_{ik}^{mul} be the ground truth for the corresponding position and class. The loss function for dense multi-label is defined as:

$$\begin{aligned} l_{mul}(\boldsymbol{y}^{mul}, \boldsymbol{m}) = \frac{1}{IK} \sum_{i}^{I} \sum_{k}^{K} y_{ik}^{mul} \log(\frac{1}{1+e^{-m_{ik}}}) \\ + (1 - y_{ik}^{mul}) \log(\frac{e^{-m_{ik}}}{1+e^{-m_{ik}}}), \end{aligned} \tag{3.5}$$

where $y_{ik}^{mul} \in \{0, 1\}$; I and K represent the number of spatial points and classes, respectively.

Similarly, let p_{ik} be the fused output at ith position for kth class, and y_i^{seg} be the ground truth for segmentation prediction at ith position. The loss function for segmentation is defined as:

$$l_{seg}(\boldsymbol{y}^{seg}, \boldsymbol{p}) = \frac{1}{I} \sum_{i}^{I} \sum_{k}^{K} \mathbb{I}(y_i^{seg} = k) \log(\frac{e^{p_{ik}}}{\sum_j e^{p_{ij}}}), \tag{3.6}$$

where $y_i^{seg} \in \{1 \ldots K\}$.

Our goal is to minimize the objective function:

$$\min l_{seg} + \lambda(l_{mul}^{(1)} + l_{mul}^{(2)} + l_{mul}^{(3)}), \tag{3.7}$$

where λ controls the balance between the segmentation block and the dense multi-label blocks. I observe this parameter is not very sensitive. We set $\lambda = 1$ to treat each part equally.

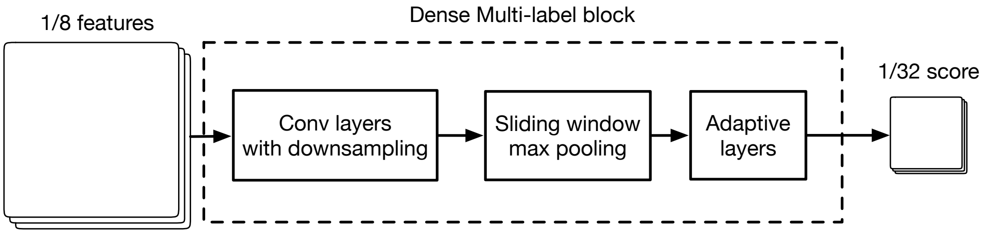


Figure 3.5: Details of a single dense multi-label block. The input features are fed into several convolutional layers and further downsampled. Then we perform sliding window with max pooling operation. After some adaptive layers, we have scores for dense multi-label at 1/32 resolution.

3.3.3 Dense Multi-label Block

The details of the dense multi-label block are shown in Figure 3.5, where the input is feature maps at 1/8 resolution, due to the downsampling in the low level feature layers. After some convolutional layers with further downsampling, the dense multi-label is performed at 1/32 resolution with the sliding window and following adaptive layers. The reason for this setting is because dense multi-label requires a large sliding window, which will become a computational burden if we work at a high resolution. Downsampling can greatly reduce the size of feature maps and more importantly, the size of sliding window will shrink accordingly, thus making the computation more efficient. On the other hand, dense multi-label requires more high level information. Therefore, working at a coarse level can capture the high level features better. The output of the dense multi-label is upsampled to be compatible with the segmentation block's output.

3.3.4 Ground Truth Generation

The ground truth for dense multi-label can be generated from the segmentation ground truth. The process is described in Figure 3.6. Firstly, the segmentation ground truth is converted to channel-wise labels, which means each channel only contains 1 or 0 to indicate whether the corresponding class appears or not. To generate a ground-truth mask for each class, for a given window size, we slide the window

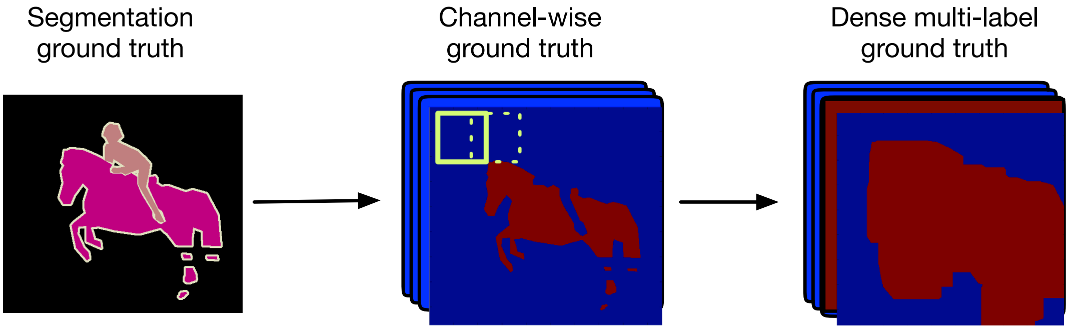

Segmentation ground truth
Channel-wise ground truth
Dense multi-label ground truth

Figure 3.6: The segmentation ground truth is firstly converted to channel-wise labels, with 0 or 1 in each channel. The ground truth for dense multi-label can be obtained by performing max pooling on the channel-wise labels.

across each binary channel and perform a max-pool operation (this is equivalent to a binary dilation using a structuring element of the same size and shape as the window). We repeat this process for each window size. As noted in section 3.3.3, the dense multi-label classification is performed at 1/32 resolution while the segmentation is at 1/8. Therefore, we generate multi-label ground-truth data at 1/8 resolution with stride 4.

3.3.5 Network Configuration

The dense multi-label module is suitable for any segmentation system and it can be easily integrated. In this study, we use Residual 50-layer network He et al. [2016] with dilated kernels Chen et al. [2015a]. In order to work at a relatively high resolution while keeping the efficiency, we use 8-stride setting, which means that the final output is at 1/8 resolution. As we mentioned in the last section, we perform dense multi-label at 1/32 resolution to make it more efficient and effective. The window sizes are then defined at 1/32 resolution. For example,let w be the window size. A window with $w = 17$ at 1/32 resolution means $4w = 68$ at 1/8 resolution. The corresponding window for the original image is $32w = 544$. We use $w_1 = 35$, $w_2 = 17$ and $w_3 = 7$ for all the experiments.

Block name	Initial layers	Stride
Low level feature block	conv1 to res3d	8
Segmentation block	res4a to res5c	1
Dense multi-label block	res4a to res5c	4

Table 3.1: Configuration for Res50 network. The low level feature block is initialized by layers "conv1" to "res3d" and has 8 stride. The segmentation block and dense multi-label blocks are initialized by layers "res4a" to "res5c" but do not share the weights with each other. The segmentation block does not have any downsampling, but the dense multi-label blocks have further 4 stride downsampling.

Table 3.1 shows the configuration with 50-layer Residual net (Res50) as the base network. The low level feature block contains layers from "conv1" to "res3d". The segmentation block and dense multi-label blocks have layers from "res4a" to "res5c" as well as some adaptive layers. It is worth noting that it does not mean these blocks will share the weights even though they initialize the weights from the same layers. After initialization, they will learn their own features separately.

3.4 Experiments and Analysis

We evaluate our model on 4 commonly used semantic segmentation datasets: ADE 20k, NYUDv2, SUN-RGBD and PASCAL-Context. Our comprehensive experiments show that dense multi-label can successfully suppress many unlikely labels, retain region consistency and thus improve the performance of semantic segmentation.

The results are evaluated using the Intersection-over-Union (IoU) score Everingham et al. [2010]. Moreover, since our original motivation is to suppress noisy and unreasonable labels to keep labels consistent with the region, we also introduce new measurements to evaluate the number of classes that are not in ground truth, and further, the number of pixels that are predicted to be these wrong classes for each image.

We only use Res50 as base network to compare and analyse the performance. For all the experiments, we use batch size of 8, momentum of 0.9 and weight decay of 0.0005.

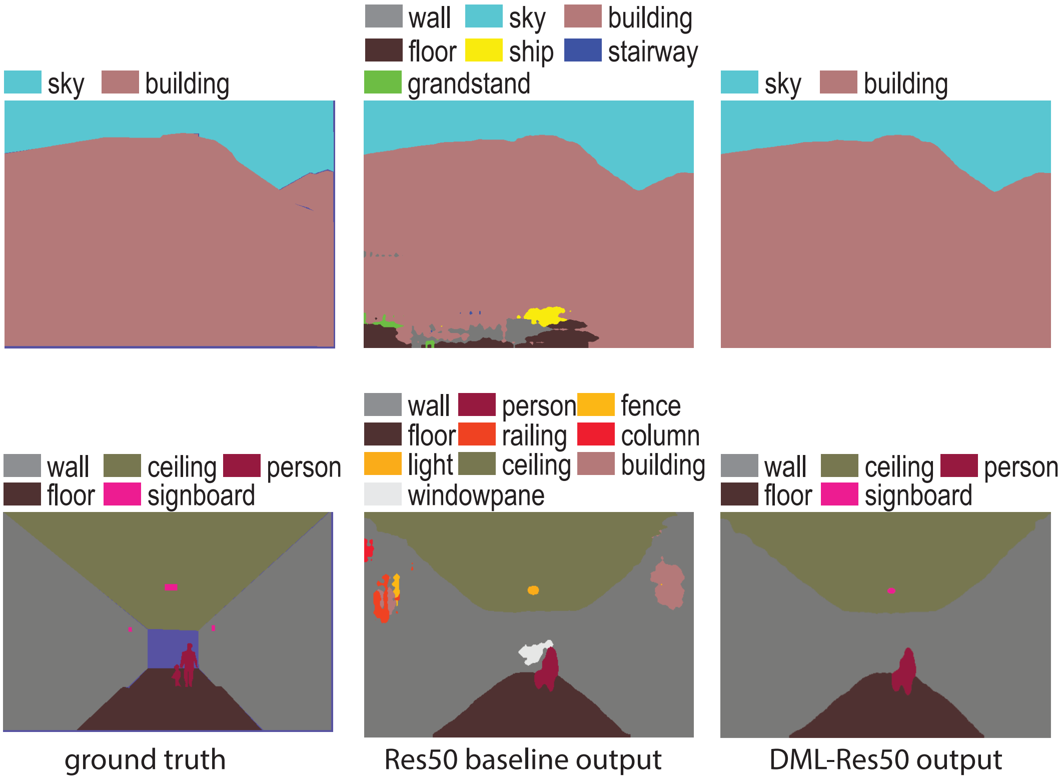


Figure 3.7: Example outputs of Res50 baseline and DML-Res50 on ADE 20k dataset.

3.4.1 Results on ADE 20k dataset

We first evaluate our result on ADE 20k dataset Zhou et al. [2016b], which contains 150 semantic categories including objects such as person, car etc., and "stuff" such as sky, road etc. There are 20210 images in the training set and 2000 images in the validation set.

As shown in Table 3.2, the model with dense multi-label (DML-Res50) yields a 2% improvement. To analyse the effectiveness of label suppression, we also use two criteria to evaluate this performance, which are shown as "Wrong class" and "Wrong labels". Wrong class means the number classes that are not supposed to appear but are mistakenly predicted by the model. Wrong labels describe how many pixels are assigned with those wrong classes. We observe that using Dense multi-

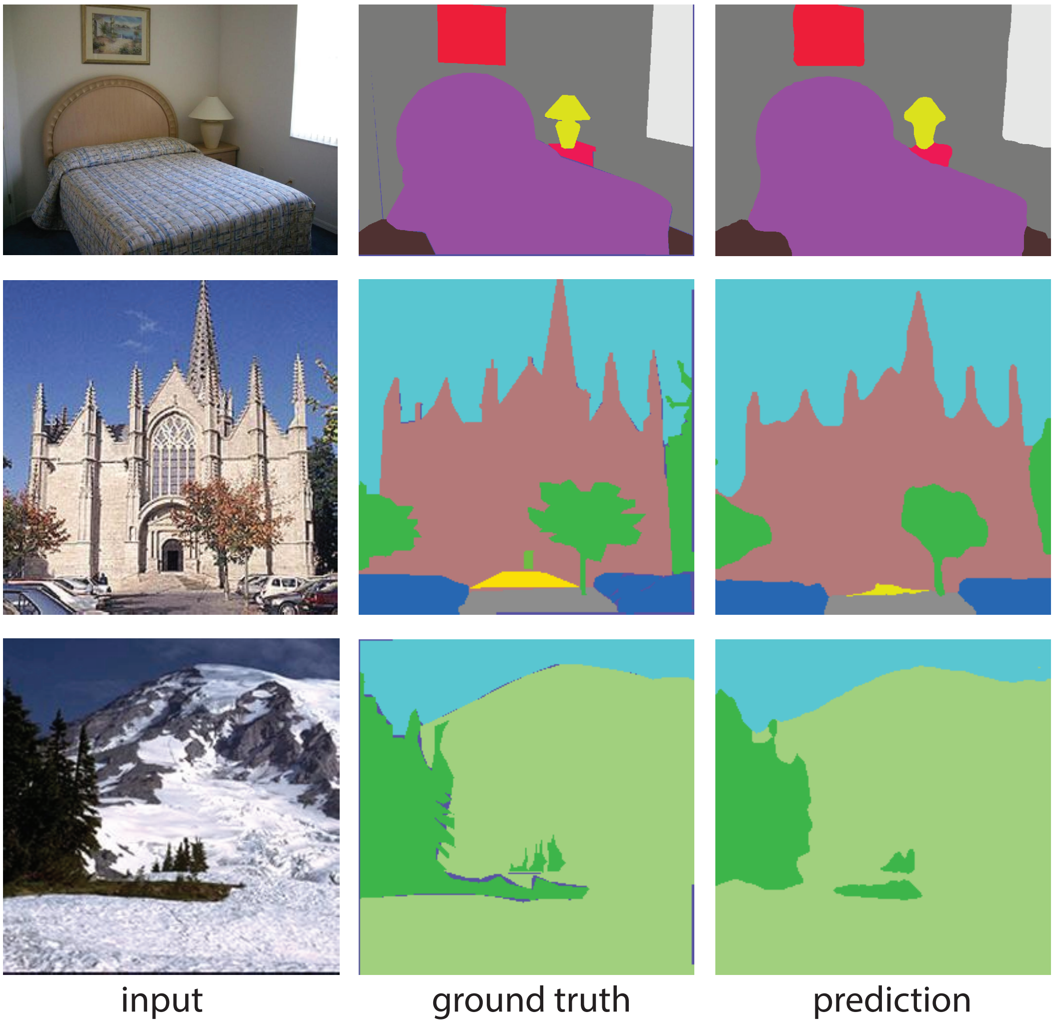


Figure 3.8: More example outputs of dense multi-label network on ADE dataset.

label effectively reduces the wrong classes and labels, by 35% and 16% respectively. Some examples are shown in Figure 3.7. To make fair comparison, all the images are raw outputs directly from the network. The last column shows the outputs from the network with dense multi-label where we can observe great scene consistency compared with the output of the baseline network shown in the middle.

In comparison with other methods, we achieve better results than the models reported in Zhou et al. [2016b], as shown in Table 3.3. More examples can be found in Figure 3.8

Model	IOU	#Wrong class	#Wrong label
Res50 baseline	34.5	5.576	21836
DML Res50	**36.49**	**3.6**	**18294**

Table 3.2: Results on ADE dataset. The dense multi-label boosts the performance by 2% of IOU and helps reduce the number of wrong class and label by 35% and 16% respectively.

Model	IOU
DilatedNet Zhou et al. [2016b]	32.31
Cascade-DilatedNet Zhou et al. [2016b]	34.90
DML-Res50(ours)	**36.49**

Table 3.3: Comparsion with other models on ADE dataset. Our model achieves the best performance.

3.4.2 Results on PASCAL-Context

PASCAL-Context dataset Mottaghi et al. [2014] is a set of additional annotations for PASCAL VOC 2010, which provides annotations for the whole scene with 60 classes (59 classes and a background class). It contains 4998 images in training set and 5105 images in validation set.

Figure 3.9 shows some typical examples on this dataset. We can also see clear scene consistency with dense multi-label involved. The outputs in the middle contain many noisy classes, especially the lower middle image contains “bird” and “sky”, which are very unlikely in this scene. From Table 3.4, we can also see the great boost with dense multi-label. The wrong classes and labels are greatly reduced by 37% and 15%.

To compare with other models, we list several results on this dataset. Since different models have various settings such as multi-scale training, extra data, etc. we also explain it in Table 3.5. Considering all the factors involved, our method is comparable since we only use Res50 as the base network and do not use mult-scale training and extra MS-COCO data for pretraining. More examples are shown in 3.10.

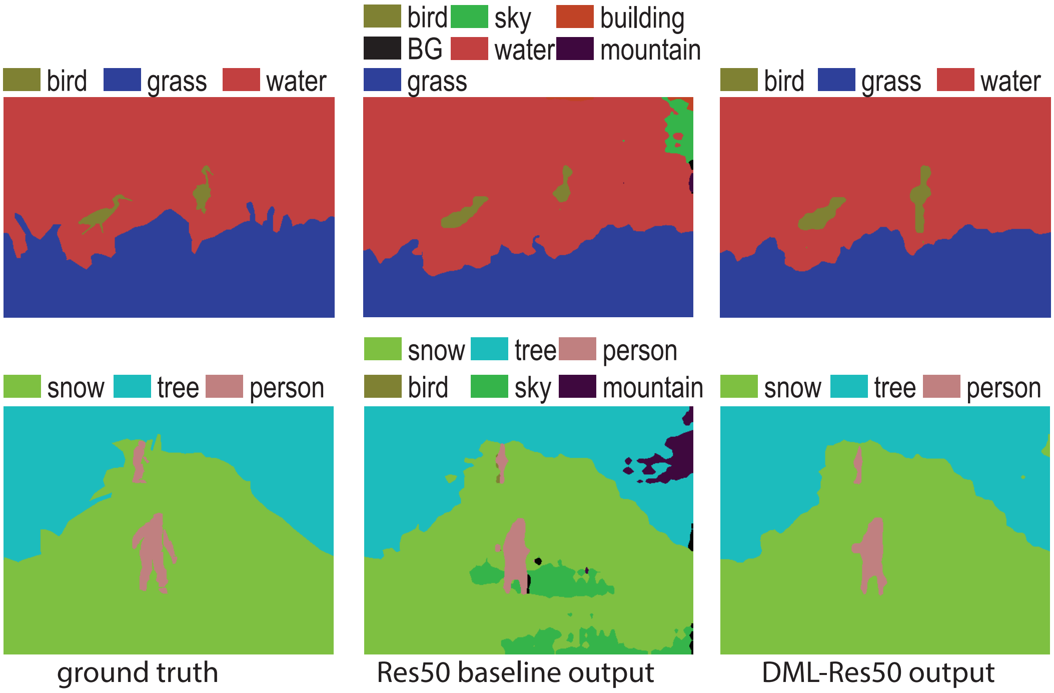


Figure 3.9: Example outputs of Res50 baseline and DML-Res50 on PASCAL-Context dataset.

Model	IOU	#Wrong class	#Wrong label
Res50 baseline	41.37	4.5	26308
DML-Res50	**44.39**	**2.8**	**22367**

Table 3.4: Results on PASCAL-Context dataset. The dense multi-label model increases the IOU by 3% and reduces the wrong classes and labels by 37% and 15%.

3.4.3 Results on NYUDv2

NYUDv2 Silberman et al. [2012] is comprised of 1449 images from a variety of indoor scenes. We use the standard split of 795 training images and 654 testing images.

Table 3.6 shows the results on this dataset. With dense multi-label, the performance is improved by more than 1%, and the number of wrong class and label decrease by about 40% and 16%. Some examples are shown in Figure 3.11. Scene consistency still plays an important role in removing those noisy labels. Compared with some other models, we achieve the best result, as shown in Table 3.7.

Model	Base	MS	Ex data	IOU
FCN-8s Long et al. [2015]	VGG16	no	no	37.8
PaserNet Liu et al. [2015]	VGG16	no	no	40.4
HO_CRF Arnab et al. [2015]	VGG16	no	no	41.3
Context Lin et al. [2017b]	VGG16	yes	no	43.3
VeryDeep Wu et al. [2016]	Res101	no	no	44.5
DeepLab Chen et al. [2016]	Res101	yes	COCO	**45.7**
DML-Res50 (ours)	Res50	no	no	44.39

Table 3.5: Results on PASCAL-Context dataset. MS means using multi-scale inputs and fusing the results in training. Ex data stands for using extra data such as MS-COCO Lin et al. [2014]. Compared with state of the art, since we only use Res50 instead of Res101 and do not use multi-scale training as well as extra data, our result is comparable.

Model	IOU	#Wrong class	#Wrong label
Res50 baseline	38.8	8.2	27577
DML-Res50	**40.23**	**4.9**	**23057**

Table 3.6: Results on NYUDv2 dataset. Dense multi-label network has 1.4% higher IOU and 40% and 16% lower wrong classes and labels respectively.

3.4.4 Results on SUN-RGBD

SUN-RGBD Song et al. [2015] is an extension of NYUDv2 Silberman et al. [2012], which contains 5285 training images and 5050 validation images, and provides pixel labelling masks for 37 classes.

Figure 3.12 shows some output comparison on this dataset, where we can easily observe the effect of dense multi-label. The results are shown in Table 3.8. The network with dense multi-label helps improve the IOU by more than 3%. The wrong classes and wrong labels also get decreased by 36% and 18% respectively. Compared with other methods, the network with dense multi-label reaches the best result, as shown in Table 3.9. More examples can be found in Figure 3.13.

3.4.5 Ablation Study on PASCAL-Context

Table 3.10 shows an ablation study on the PASCAL-Context. The Res50 baseline yields mean IOU of 41.37%. Treating this as a baseline, we introduce dense multi-

Model	IOU
FCN-32s Long et al. [2015]	29.2
FCN-HHA Long et al. [2015]	34.0
Context Lin et al. [2017b]	40.0
DML-Res50 (ours)	40.23

Table 3.7: Comparison with other models on NYUDv2 dataset. Our method achieves the best result.

Model	IOU	#Wrong class	#Wrong label
Res50 baseline	39.28	5.3	24602
DML-Res50	**42.34**	**3.36**	**20104**

Table 3.8: Results on SUN-RGBD dataset. Dense multi-label helps increase the performance by more than 3% of IOU and decrease the wrong classes and labels by 36% and 18%.

level module. Firstly, in the one level setting, we use the largest window size, which is basically global multi-label classification. Accordding to the results, the first level gives the biggest boost. With 2 levels involved, the global and mid-level window, the performance is improved further. The final level, the smallest window, brings 0.6% more improvement. The dense multi-label module helps improve the performance by 2.2% in total. After using CRF as post-processing, we can achieve IOU of 44.39 without using extra MS COCO dataset.

3.4.6 Failure Analysis

We also observed some failure cases from the outputs, with two main types of failure shown in Figure 3.14. The left half of Figure 3.14 depicts a failure mode in which the objects are totally misclassified into another class; here the assigned lables are consistent due to the dense multi-label module but the object/region class is wrong. Another failure type is shown in the right half of the figure, where the labels are consistent but the model failed to detect some objects or detected some non-existing objects. In the former case, the error here appears primarily to be one exacerbated by the dense multi-label prediction. This could be mitigated by improving the quality of dense multi-label prediction and/or adjusting the balance between the dense multi-

Model	IOU
Kendall Kendall et al. [2015]	30.7
Context Lin et al. [2017b]	42.3
DML-Res50 (ours)	42.34

Table 3.9: Comparison with other models on SUN-RGBD dataset. We achieve the best result with dense multi-label network.

Model	IOU
Res50 baseline	41.37
DML-Res50 1level	42.52
DML-Res50 2level	42.95
DML-Res50 3level	43.59
DML-Res50 3level + CRF	**44.39**

Table 3.10: Ablation study on PASCAL-Context.

label module and the segmentation part. We emphasize however, that the dense multi-label technically can be integrated into any segmentation system to help retain the consistency, and our results show the efficacy of doing so.

3.5 Conclusion

In this study, we propose a dense multi-label module to address the problem of scene consistency. With comprehensive experiments, we have shown that dense multi-label can enforce the scene consistency in a simple and effective way. More importantly, the dense multi-label is a module and can be easily integrated into other semantic segmentation systems.

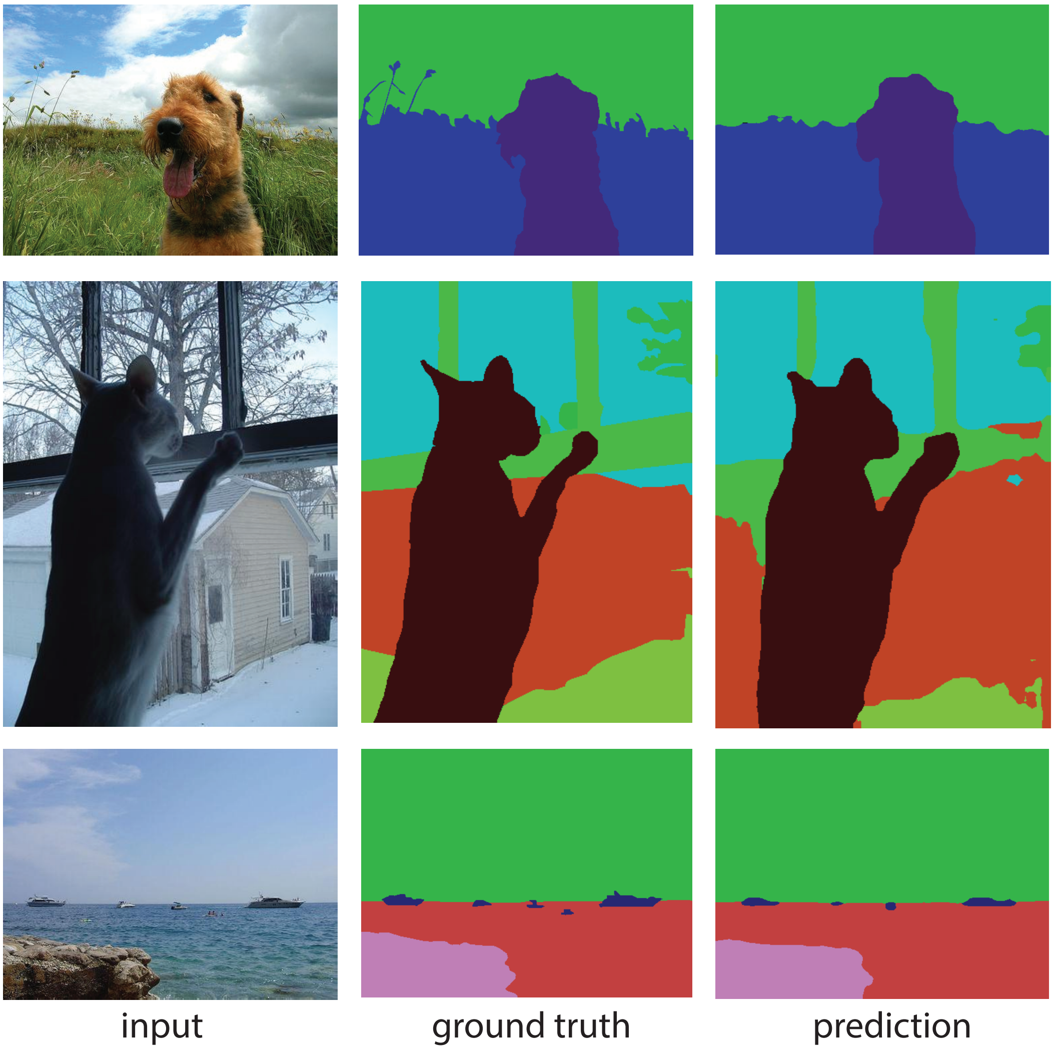
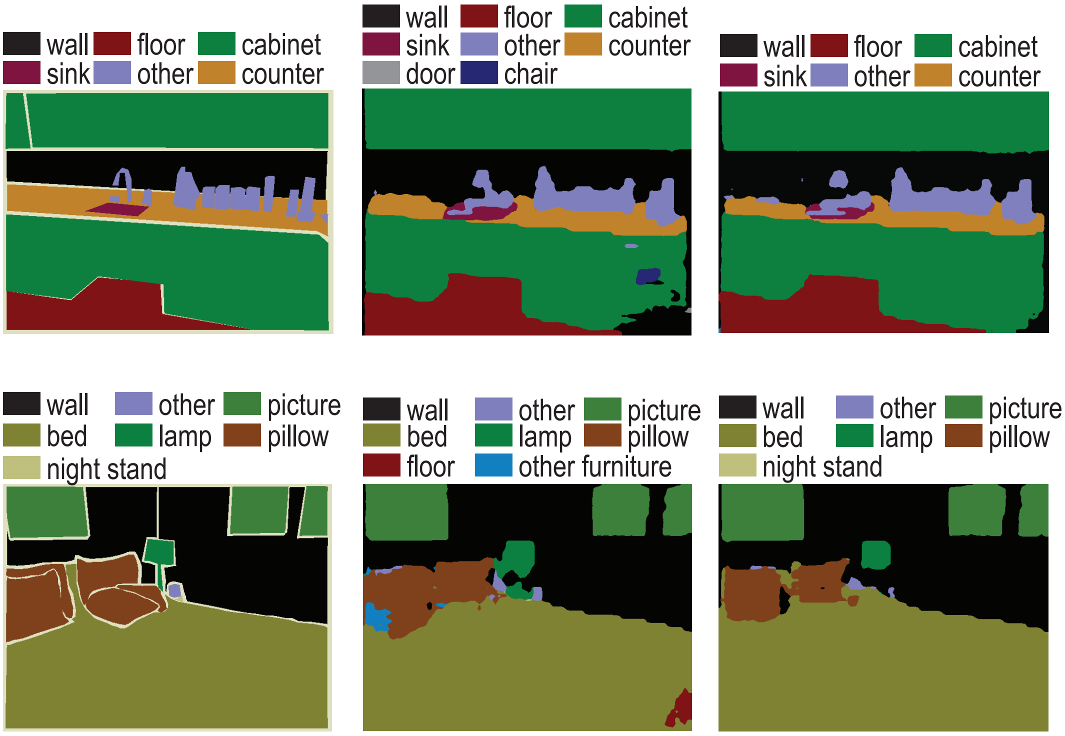


Figure 3.10: More example outputs of dense multi-label network on PASCAL-Context dataset.



ground truth Res50 baseline output DML-Res50 output

Figure 3.11: Example outputs of Res50 baseline and DML-Res50 on NYUDv2 dataset.

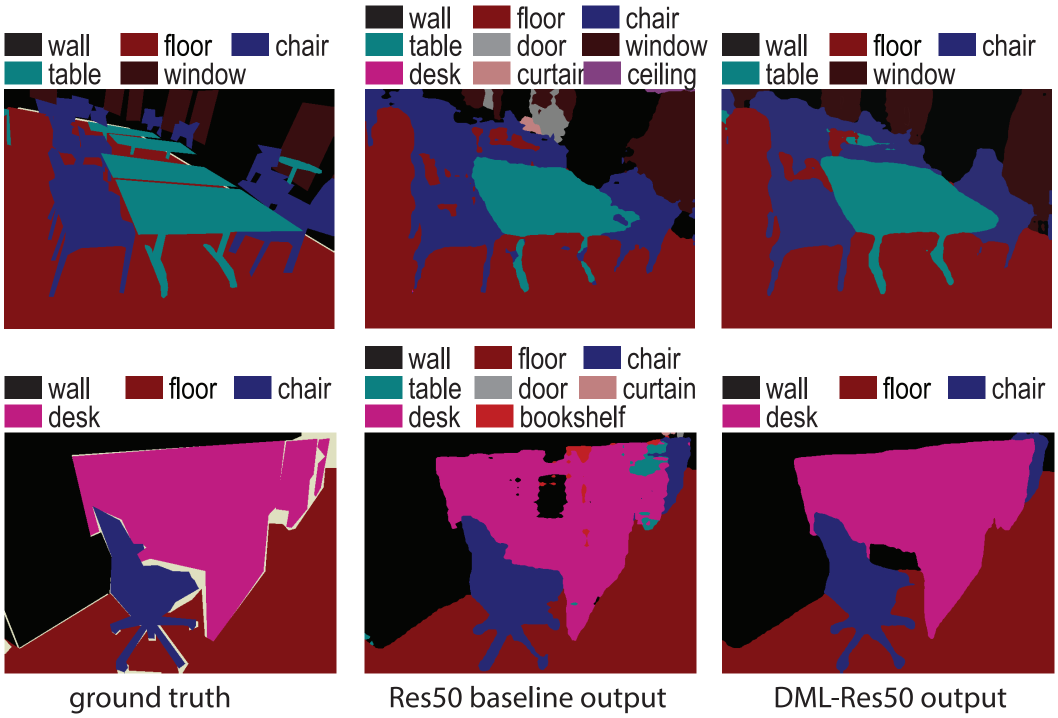


Figure 3.12: Example outputs of baseline Res50 and DML-Res50 on SUN-RGBD dataset.

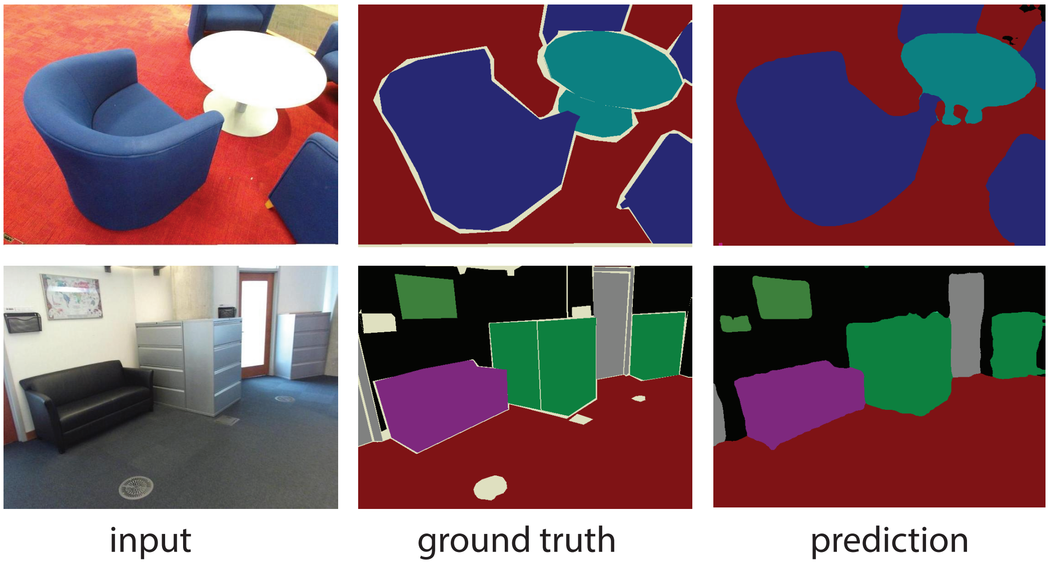


Figure 3.13: Good examples on SUN-RGBD dataset.

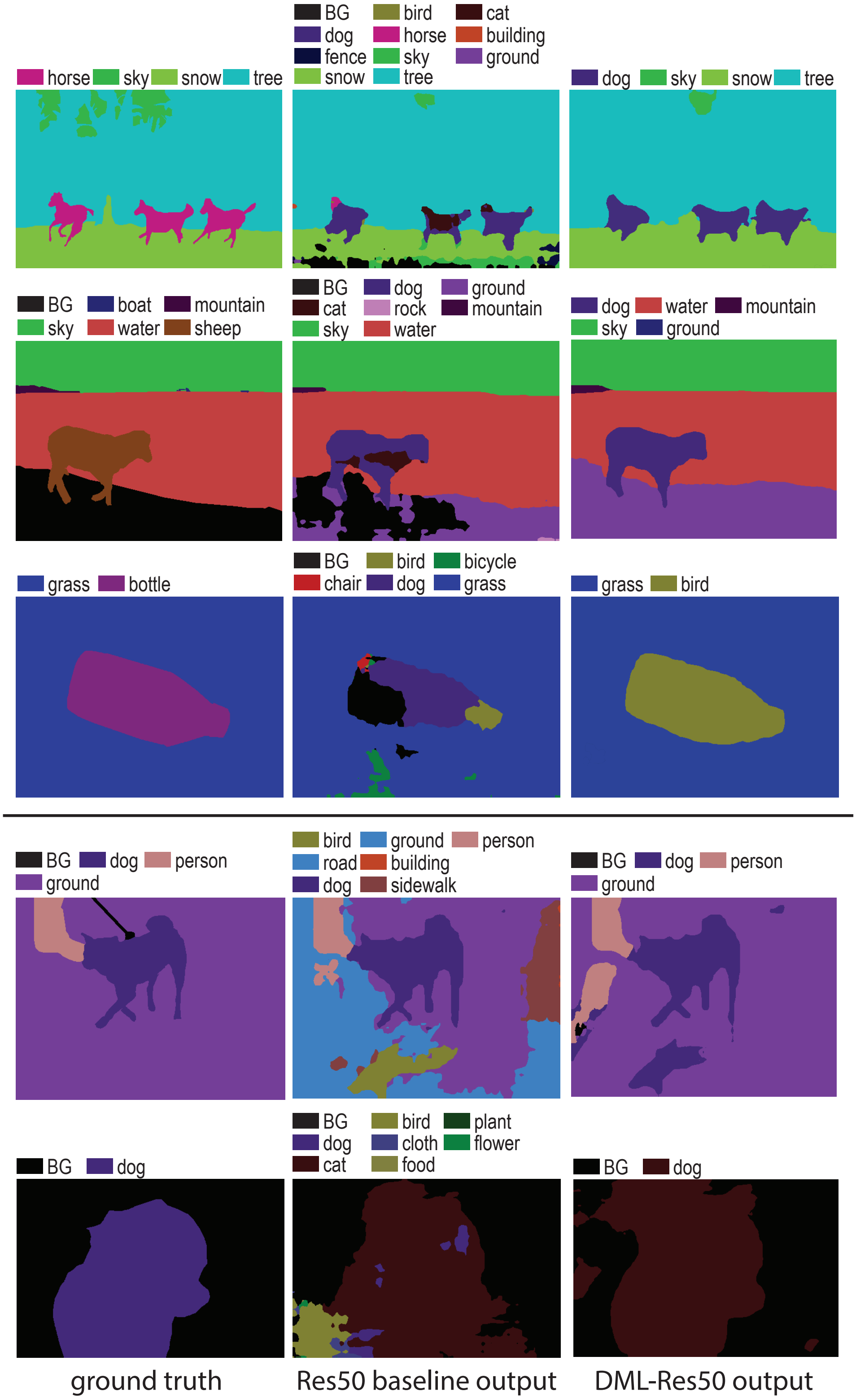


Figure 3.14: Examples of failed case.

Chapter 4

Weakly Supervised Segmentation Based on Web Image Co-segmentation

4.1 Introduction

In the previous chapter, we have discussed an important problem in semantic segmentation, scene consistency, and proposed a dense multi-label module to impose such consistency constraint. In this chapter, we will discuss another problem, weakly supervised learning. We notice that in a classification task (Krizhevsky et al. [2012]; Simonyan and Zisserman [2015]; He et al. [2016]), labels are simply image-level labels. However, dense prediction tasks such as semantic segmentation require pixel-label labels, which involve great annotation effort. As pointed in Bearman et al. [2016], it takes averagely 239.7 seconds to process one image to get full supervision, and only 20 seconds to get image-level labels. Obviously, reducing the labelling time makes us achieve training data more cheaply, and further enable more training data to be collected easily.

To reduce the cost of generating pixel-level ground truth, it is worth exploring methods working in weakly and semi-supervised settings. In addition to image-level labels (Kolesnikov and Lampert [2016]; Pathak et al. [2015]), other means of supervision have also been utilized such as bounding boxes Pathak et al. [2015], scribble Lin et al. [2016a], points Bearman et al. [2016] etc. Point supervision is able to indicate the location of objects, and bounding boxes and scribble can even indicate the location as well as the extent of objects. Image-level supervision is obviously the most challenging task where we only know the existence of objects. The objective of this work is to perform semantic segmentation only with image-level labels and try to reduce the gap between weakly supervised methods and fully supervised methods.

For weakly supervised methods with only image-level labels (Bearman et al. [2016]; Wei et al. [2017b]; Pinheiro and Collobert [2015]), it is very common to introduce some prior knowledge by other auxiliary methods such as Objectness Alexe et al. [2012], Multiscale Combinatorial Grouping (MCG) Arbeláez et al. [2014], and saliency dectection Jiang et al. [2013]. It is worth noting that these methods might rely on more than just image-level labels to be trained. For example, Objectness Alexe et al. [2012] requires bounding boxes to train and MCG requires pixel-level ground truth. Therefore, strictly speaking, using only image-level labels indicates no more supervision other than image-level labels including the training of the other methods involved.

In this chapter, we aim to propose a weakly supervised framework for semantic segmentation that strictly complies with the rules of image-level supervision, which means apart from image-level labels, there is no extra supervision implicitly involved. Our method is based on a co-segmentation method Chen et al. [2014], which is an unsupervised and robust method for large scale co-segmentation. More specifically, our framework has two steps, training an initial network as a mask generator and training another network as the final model. The mask generator is trained with the retrieved images from the internet and is used to provide masks for the final model. The final model is trained by the masks from the mask generator as well as the image-level labels provided. In the first step, we first retrieve images from search engines (Flickr and Google) according to class names. Then for each class, which has a large number of images containing the same semantic object, we use co-segmentation to extract masks for each image. These masks are used to train the mask generator. In the second step, the mask generator produces masks for the target dataset, e.g. PASCAL VOC 2012. With these mask as well as the image-level labels that helps to eliminate impossible predictions, we are able to train the final model with high quality ground truth.

Our contributions are as follows:

- We propose a new weakly supervised method based on co-segmentation. Apart from image-level supervision, there is no extra supervision involved. We show that this two-step framework is simple but effective.

- We use the most popular benchmark dataset, PASCAL-VOC12 to demonstrate the performance of our framework and we achieved state of the art performance.

4.2 Background

4.2.1 Weakly Supervised Semantic Segmentation

In the literature, there have been many weakly- and semi-supervised methods proposed for semantic segmentation. Those methods utilize different forms of supervision including image-level labels (Pathak et al. [2015]; Kolesnikov and Lampert [2016]; Oh et al. [2017]; Wei et al. [2017b]; Pinheiro and Collobert [2015]), bounding boxes Dai et al. [2015], points Bearman et al. [2016], scribble Lin et al. [2016a], or combined supervision (Hong et al. [2015]; Papandreou et al. [2015]).

Our work only uses image-level labels. Therefore, we will discuss some works with the same supervision setting. Pinheiro and Collobert [2015] treat weakly supervised segmentation as a Multiple Instance Learning (MIL) task. They claim that using image-level label for training makes the model learn to discriminate the right pixels, and using some extra smoothing priors can give good pixel labelling results. Pathak et al. [2015] introduce a constrained CNN for weakly supervised training by setting a series constraints for object size, foreground, background. Kolesnikov and Lampert [2016] propose a "seed, expand and constrain" framework where they employ localization cues from DCNN to find the object's location, use global weighted rank pooling to expand the mask, and use Conditional Random Fields (CRFs) to refine the boundary. Oh et al. [2017] combine localization cues and saliency to localize objects and obtain their extent so as to generate semantic masks. Wei et al. [2017b] use a simple to complex framework for weakly supervised learning. They retrieve images from Flickr and use saliency maps as ground truth for training. Then they train the model in a "simple to complex" fashion as they train initial model, enhanced model and powerful model step by step. It is worth noting that using saliency has its own limitation because it is class-agnostic and the interesting region might not be the salient region detected. Our co-segmentation based framework will be more robust and respect the co-occurrence regions. More details will be discussed in Section 4.3.1.

4.2.2 Co-segmentation

The objective of co-segmentation is to segment similar objects from a pair of images or a group of images. Some methods are focused on small scale co-segmentation where only a small number of images are involved (Hochbaum and Singh [2009]; Joulin et al. [2010]; Dai et al. [2013]). These bottom-up methods use low level features to find similarities between images and formulate optimization problems to find the co-segments.

There are some other methods aiming at large scale co-segmentation with a large number of images presented including noisy data (Faktor and Irani [2013]; Chen et al. [2014]). Faktor and Irani [2013] define co-segmentation as a composition problem where good co-segments can be easily composed and vice versa. The method is suitable for both large scale data and even a single image. Chen et al. [2014] propose an approach to combine top-down segmentation priors learned from visual subcategories and bottom-up cues to produce good co-segmentation from noisy web data. Our framework is based on this technique, which can provide us good training masks for retrieved images.

4.3 Method

Figure 4.1 shows the overall pipeline of our framework. Basically, there are two steps. In the first step, we retrieve images according to class names, e.g. 20 object names in PASCAL-VOC12, and we use large scale co-segmentation method Chen et al. [2014] for each class group. In each group, we assume that most of the images contain the desired object in the corresponding class. Since the co-segmentation method has great ability to tolerate noise, we can control the balance between recall and precision and get high quality results. After we obtain the co-segmentation results, we treat them as ground truth masks and train a CNN as initial mask generator. In the second step, we use the mask generator to produce masks for images of the target dataset, e.g. PASCAL VOC 2012. Furthermore, since we have access to image-level labels, we can use it to eliminate impossible predictions and enhance the masks. Finally we use these high quality masks as well as image-level labels to train another CNN as the final model. More details will be discussed in later sections.

Let $\mathbf{I}$ be an image, and $\mathbf{Y} = [y_1, y_2, ..., y_C]$ be the label vector, where $y_j = 1$ if the image is annotated with class j and otherwise $y_j = 0$; C is the number of class. Assume we have training data, $D = \{(\mathbf{X}_n, \mathbf{Y}_n)\}_{n=1}^{N}$, where N is the number of training data. Our objective is to learn a DCNN function $f(\mathbf{X};`)$ parameterized by `.

4.3.1 Saliency vs Co-segmentation

In this section, let us first discuss the advantages of co-segmentation methods in terms of estimating the mask of retrieved images compared with saliency detection. In Wei et al. [2017b], saliency detection method DRFI Jiang et al. [2013] is used to generate masks for internet retrieved images. Generally speaking, saliency detection frameworks require relatively clean background and high contrast between foreground and background. For relatively complex images, saliency has its own limitation, which may detect undesired objects. One situation could be that the salient part is not the target object. Another situation could be that saliency only concentrates on the most salient part so that the other parts of the object are ignored.

In contrast, co-segmentation has more advantages in terms of localizing the desire object and finding the accurate contour. In Chen et al. [2014], to perform co-segmentation on a group of images, e.g. a group of car images, it first uses low-level features to find aligned homogeneous clusters and learns a set of visual subcategories. Then top-down segmentation priors can be created and used to extract the final segment for each image with the help of graph-cut algorithm. This bottom-up and top-down procedure can better localize the common objects in the image collection and obtain accurate contour.

Figure 4.2 shows some examples of saliency map produced by DRFI and co-segmentation mask. In each half, the first column is the original image; the middle is the saliency map and the last is the co-segmentation mask. In the left half, we show some example when saliency fails to detect the object as a whole part. Instead only a small portion has high probability, which makes it difficult to capture the entire object. In comparison, co-segmentation can capture the whole part. In the right half, there are some circumstances where saliency produces lots of false positives and fails to detect the part we expect. Co-segmentation can find better patterns from the large collection of images containing the common objects, which gives us more promising results.

4.3.2 Training Initial Mask Generator

To train the initial mask generator, we retrieve images from Flickr and Google and use co-segmentation to obtain estimated ground truth for each image. We define $\mathbf{M}$ as the mask, and this data set is expressed as $D_1 = \{(\mathbf{X}_n, \mathbf{M}_n)\}_{n=1}^{N_1}$, where N_1 is the number of retrieved images. We train the initial Mask generator as in standard fully-supervised framework using softmax loss. For a single image, the loss is defined as:

$$\mathcal{L}_1 = \frac{1}{I}\sum_i^I\sum_j^C \mathbb{1}(m_i = j)\log(\frac{\exp(f_{ij})}{\sum_k \exp(f_{ik})}), \tag{4.1}$$

where f_ij is the prediction from the network for class j at spatial position i; m_i is the training mask for spatial position i; I is the number of spatial points.

The network structure we use is Resnet50 He et al. [2016] with dilations, similar to DeepLab Chen et al. [2016].

4.3.3 Training Final Model

After we trained the initial mask generator, we can use it to generate pixel-level labels for images of the target dataset. However, these labels could be noisy. For example, "cow" pixels could be predicted to "sheep" pixels but the image does not have sheep. Fortunately since image-level labels are accessible to these images, we can use it as constraints to eliminate impossible predictions to obtain high quality masks. For an image in the target dataset, we apply the following operation:

$$m_i = \underset{j\in\{0,..,C-1\}}{\arg\max}\, y_j f_{ij}, \tag{4.2}$$

where $y_j = 1$ if class j is shown and otherwise 0; f_{ij} is the score for class j at position i; m_i is the label for position i.

So far, we have the masks for all the images in the target dataset. The training set can be expressed as $D = \{(\mathbf{X}_n, \mathbf{Y}_n, \mathbf{M}_n)\}_{n=1}^N$.

We train the final model also in fully supervised setting. Furthermore, in order to include more context information, we borrow the idea from Shen et al. [2017b] where multi-label classification is combined to enforce scene consistency. For simplicity, we only add one branch for global context. Thus the training process is guided by two losses, softmax loss and multi-label loss. The softmax loss is the same as in Equation

4.1, while the multi-label loss for single image is a binary logistic loss expressed as:

$$\mathcal{L}_2 = \frac{1}{C}\sum_j y_j \log(\frac{1}{1+e^{-p_j}}) + (1-y_j)\log(\frac{e^{-p_j}}{1+e^{-p_j}}), \tag{4.3}$$

where p_j is the prediction for class j.

The objective is to minimize the combined loss for all images:

$$\mathcal{L} = \sum_n^N \mathcal{L}_1^n + \lambda \mathcal{L}_2^n, \tag{4.4}$$

where λ controls balance between losses.

Intuitively, multi-label classification imposes constraints to the segmentation prediction so that the prediction would be consistent to the scene globally. Implausible predictions will be largely modified. The final model still uses Resnet50 with dilations as basic structure. The difference is the multi-label branch. The branch shares some low-level layers with the main branch and has its own high level layers.

4.4 Experiments

4.4.1 Datasets

Retrieved Dataset: We construct a dataset by retrieving images from Flickr and Google. By using the corresponding class names in our target dataset PASCAL VOC 2012 as queries, we retrieve 75802 images in total. It is worth noting that we do not have any filtering process as in Wei et al. [2017b] where they adopt multiple methods to filter the crawled images. Since the co-segmentation we use has good tolerance to noise, we simply use all the images crawled. The only preprocessing is resizing all the images so that the maximum dimension is 340. This dataset is used to train our initial mask generator.

PASCAL VOC 2012: We train and evaluate the final model on this dataset. The original dataset Everingham et al. [2010] contains 1464 training images, 1449 validation images and 1456 testing images. The dataset is further augmented to 10582 training images as in Hariharan et al. [2011]. There are totally 21 semantic classes in the benchmark. The evaluation metric is the standard Intersection over Union (IoU) averaged on all 21 classes.

4.4.2 Experiment Setup

Co-segmentation: We use the public available code of Chen et al. [2014]. For all the retrieved images, the images belonging to the same semantic class will be in a large group for co-segmentation. Therefore, there are 20 groups (excluding background class) of images. We used all default settings without tweaking any parameters. After obtaining the results, we only keep the images that have foreground pixels between 20%-%80 of the whole image, which results in 37211 images.

Initial mask generator: We use the images from co-segmentation to train the mask generator. To make the images compatible to PASCAL VOC 2012, we resize the original images and masks so that the maximum dimension is 500. The network is trained with public toolbox MXNet Chen et al. [2015b]. The network structure is basically Resnet50 with dilated convolutions, which has resolution of 1/8, similar to Deeplab structure Chen et al. [2016]. We use standard Stochastic Gradient Descent (SGD) with batch size 16, crop size 320, learning rate 16e-4, weight decay 5e-4 and momentum 0.9. The training takes around 5000 iterations. The learning rate is decreased by factor of 10 once for further fine-tuning. It is worth noting that longer training time will worsen the performance because the model might fit to some noisy data or bad quality data.

Final model: After having the mask generator trained, we use it to generate masks for 10582 training images in PASCAl VOC 2012. Following Equation 4.2, we make predictions with the image-level label constraint. Also we use multi-scale inference to combine results at different scales to increase the performance, which is common practice as in Lin et al. [2017a]; Peng et al. [2017]. The final model is trained only using these data. The network structure is a little different from the initial mask generator. As mentioned in Section 4.3.3, we include a global multi-label branch to enforce the scene consistency globally, it turns out to be very useful, as shown in later ablation study in Section 4.4.4. The parameters are similar to the training process for the mask generator. The extra parameter λ in Equation 4.4 is set to 1.0. Besides, we train the model for 11000 iterations and further fine-tune it for several iterations with the learning rate decreased by factor of 10. The final outputs are post-processed by CRF Krähenbühl and Koltun [2011].

4.4.3 Experimental Results

Table 4.1 and 4.2 show the IoU scores of our method and other weakly-supervised methods on validation set and test set of PASCAL VOC 2012, where we achieve IoU of 56.9[1] on test set. From the table, we can easily see that our method outperforms other methods in majority of classes. For some classes, our method increases the score by a big margin, for example, bottle, cow, bus. This big improvement benefits from the good quality masks produced by co-segmentation. For some low score classes, such as chair and person, we suspect it is due to the extreme co-occurrence, for which co-segmentation always treats them as a whole part. For instance, chair always shows with dining-table or motorbike always appears with person. Some examples can be seen in later failure analysis in Section 4.4.5.

We also compare the results with other methods with stronger supervision, as shown Table 4.3. In the upper half, those are methods that either use stronger supervision apart from image-level labels or use other techniques that involve other supervisions. In Lin et al. [2016a] and Dai et al. [2015], their supervision, scribble and bounding box can indicate not only the location of objects but also the extent of objects, which is far more informative than image-level labels. In Hong et al. [2015], they use pixel-level masks, which, even with relatively small number, can greatly

[1] http://host.robots.ox.ac.uk:8080/anonymous/NNRJCF.html

Method	bk	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mean
EM-Adapt	67.2	29.2	17.6	28.6	22.2	29.6	47.0	44.0	44.2	14.6	35.1	24.9	41.0	34.8	41.6	32.1	24.8	37.4	24.0	38.1	31.6	33.8
CCNN	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
MIL+seg	79.6	50.2	21.6	40.9	34.9	40.5	45.9	51.5	60.6	12.6	51.2	11.6	56.8	52.9	44.8	42.7	31.2	55.4	21.5	38.8	36.9	42.0
SEC	82.4	62.9	26.4	61.6	27.6	38.1	66.6	62.7	75.2	**22.1**	53.5	28.3	65.8	57.8	**62.3**	52.5	32.5	62.6	32.1	45.4	45.3	50.7
STC	84.5	**68.0**	19.5	60.5	42.5	44.8	68.4	64.0	64.8	14.5	52.0	22.8	58.0	55.3	57.8	**60.5**	40.6	56.7	23.0	57.1	31.2	49.8
WebS	84.3	65.3	**27.4**	65.4	**53.9**	46.3	70.1	**69.8**	**79.4**	13.8	61.1	17.4	**73.8**	58.1	57.8	56.2	35.7	66.5	22.0	50.1	46.2	53.4
Ours	**85.8**	53.0	24.0	**69.4**	36.7	**64.3**	**81.9**	64.6	74.5	11.4	**70.2**	**34.2**	72.7	**66.3**	60.5	42.3	**45.9**	**71.6**	**34.7**	**66.6**	**53.3**	**56.4**

Table 4.1: Results on the PASCAL VOC 2012 validation set, compared with other methods, EM-Adapt (Papandreou et al. [2015]), CCNN (Pathak et al. [2015]), MIL+seg (Pinheiro and Collobert [2015]), SEC (Kolesnikov and Lampert [2016]), STC (Wei et al. [2017b]) and WebS (Jin et al. [2017]).

Method	bk	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mean
EM-Adapt	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
CCNN	70.1	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+seg	78.7	48.0	21.2	31.1	28.4	35.1	51.4	55.5	52.8	7.8	56.2	19.9	53.8	50.3	40.0	38.6	27.8	51.8	24.7	33.3	46.3	40.6
SEC	83.5	56.4	28.5	64.1	23.6	46.5	70.6	58.5	71.3	**23.2**	54.0	28.0	68.1	62.1	70.0	55.0	38.4	58.0	39.9	38.4	**48.3**	51.7
STC	85.2	62.7	21.1	58.0	31.4	55.0	68.8	63.9	63.7	14.2	57.6	28.3	63.0	59.8	67.6	**61.7**	42.9	61.0	23.2	52.4	33.1	51.2
WebS	85.8	**66.1**	**30.0**	64.1	**47.9**	58.6	70.7	68.5	**75.2**	11.3	62.6	19.0	**75.6**	67.2	**72.8**	61.4	44.7	**71.5**	23.1	42.3	43.6	55.3
Ours	**86.9**	57.9	26.3	**65.1**	28.3	**63.9**	**80.6**	**70.7**	68.8	15.5	**67.1**	**37.3**	74.2	**70.1**	69.9	45.9	**50.6**	68.0	**43.9**	**58.7**	45.5	**56.9**

Table 4.2: Results on the PASCAL VOC 2012 test set, compared with other methods, EM-Adapt (Papandreou et al. [2015]), CCNN (Pathak et al. [2015]), MIL+seg (Pinheiro and Collobert [2015]), SEC (Kolesnikov and Lampert [2016]), STC (Wei et al. [2017b]) and WebS (Jin et al. [2017]).

boost the performance. For Oh et al. [2017] and B et al. [2016], the techniques involved, saliency and MCG, require additional supervision, e.g. bounding boxes or pixel-level masks to train. We can safely conclude that with stronger supervision, the performance gets better accordingly. Our method only uses the weakest supervision but can reach competitive performance compared with other supervision settings. Some qualitative results are shown in Figure 4.3.

4.4.4 Ablation Study

To analyse the effect of each part in the framework, we conducted an ablation experiment, whose results on validation set are shown in Table 4.4. In the first stage of the framework, the initial mask generator is trained to reach IoU of 48.3. Then if we use the same structure without the global multi-label module mentioned in Section 4.3.3, we only get IoU of 53.3. With the help of the global multi-label module, this performance can be increased to 55.1. With some post-processing, multi-scale inference and CRF, the final score is 56.4 on validation set.

4.4.5 Failure Analysis

In Figure 4.4, there are some failure cases presented. In the left half, it is a type of failure where two objects are recognized as one part due to strong co-occurrence. The right half shows another type of failure where the object is underestimated or overestimated. The reason could be some bad quality masks produced by co-segmentation.

4.5 Conclusion

We have presented a framework for weakly supervised semantic segmentation using only image-level labels. The framework utilises co-segmentation and retrieved images from the internet to obtain training data with pixel-level masks. Our two-step framework uses these high quality masks as well as the image-level labels of

Method	Val	Test	Supervision
Lin et al. [2016a]	63.1	-	Scribble supervision
Dai et al. [2015]	62.0	64.6	Bounding box supervision + MCG
Hong et al. [2015]	62.1	62.5	20 fully supervised images for each class
Oh et al. [2017]	55.7	56.7	Extra bounding box supervision for saliency detection
B et al. [2016]	54.3	55.5	MCG
Bearman et al. [2016]	46.1	-	Point supervision on each class
Pathak et al. [2015]	35.3	35.6	Image-level labels (weakest)
Kolesnikov and Lampert [2016]	50.7	51.7	
Papandreou et al. [2015]	-	39.6	
Wei et al. [2017b]	49.8	51.2	
Ours	56.4	56.9	

Table 4.3: Comparison with methods with stronger supervision. From the table, it can be safely concluded that generally stronger supervision leads to better performance. Although our method only uses image-level labels, which is the weakest supervision, we have achieved very competitive results.

Model	IoU
Initial mask generator	48.3
Simple final model	53.3
Final model with multi-label module	55.1
Final model+MS infer+CRF	**56.4**

Table 4.4: Results with different settings on validation set.

the target dataset to train a semantic segmentation network. Based on the experiments shown on a popular benchmark dataset, we show that our simple but effective framework reaches state of the art performance.

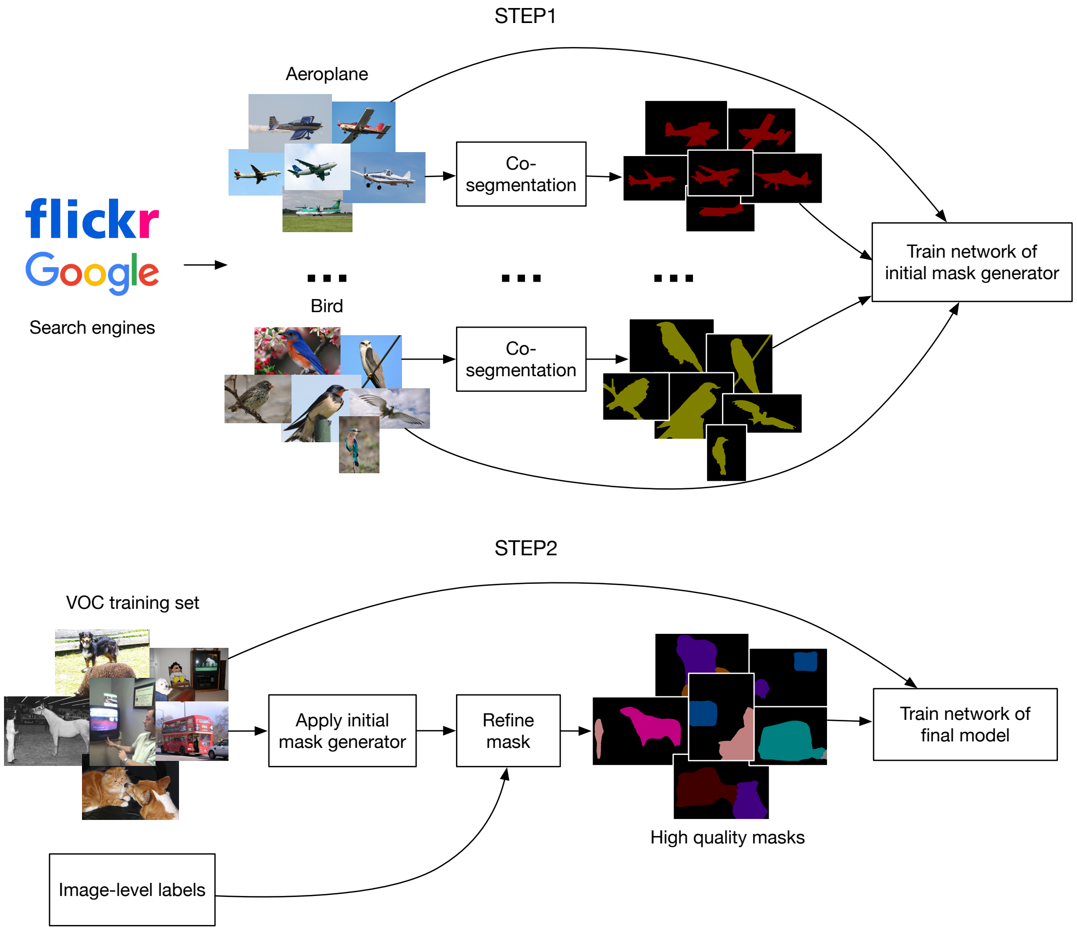


Figure 4.1: Illustration of the framework. In the first step, we firstly retrieve images from searching engines according to different semantic class names. Then co-segmentation is used to get pixel-wise masks for each semantic class group. Next we use these masks together with the images to train the initial mask generator. In the second step, we apply the mask generator to obtain pixel-wise masks for the target dataset, PASCAL VOC 2012, and further we refine the masks by using image-level labels to remove wrong predictions. The final model is trained with high quality masks.

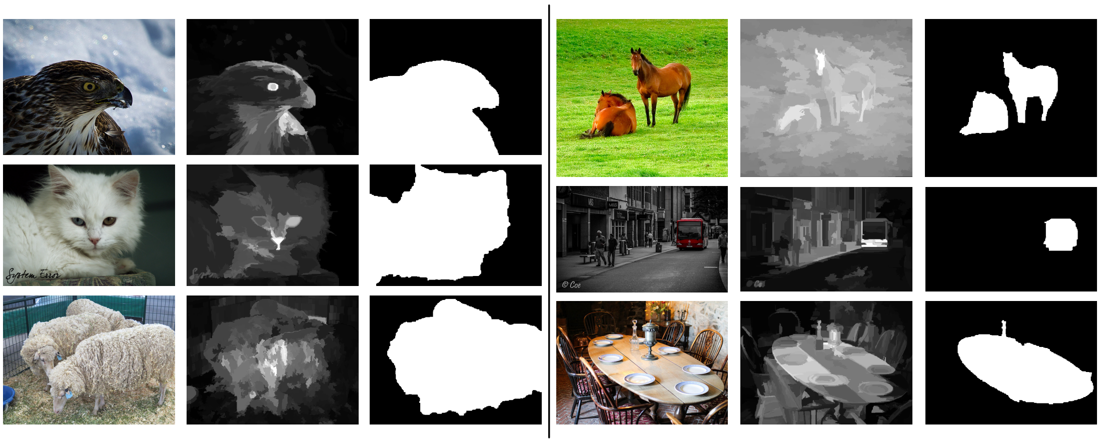


Figure 4.2: Comparison between saliency and co-segmentation. In each half, the first column is the original image, the middle is the saliency and the last is the segmentation mask. In the left half, these are examples when saliency fails to detect the object as a whole part. In the right half, it shows some cases where saliency gives high probability to many parts and fails to detect desired object.

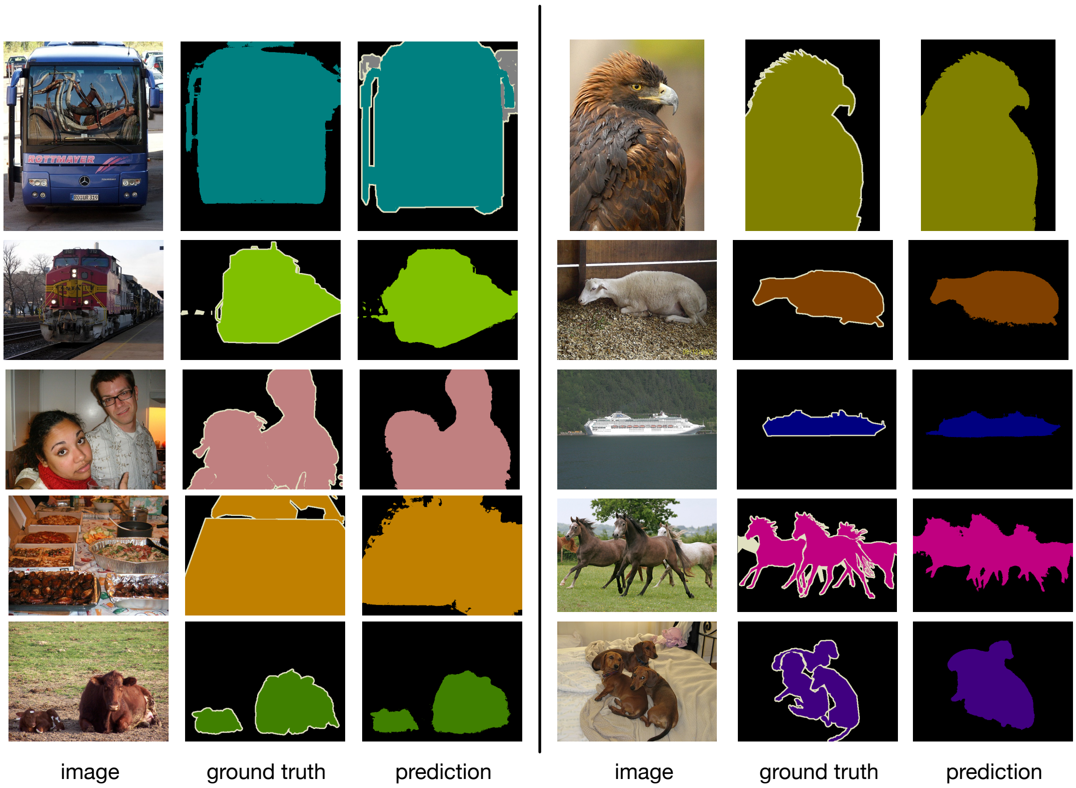


Figure 4.3: Qualitative segmentation results on PASCAL VOC 2012 validation set.

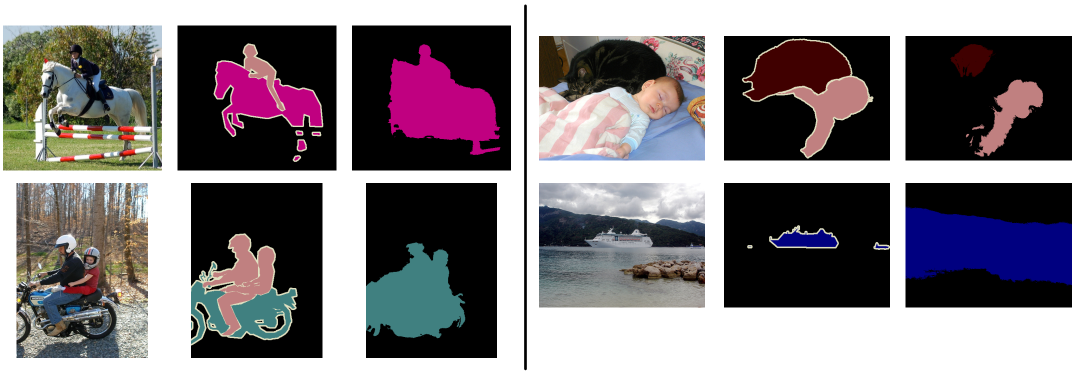


Figure 4.4: Examples of some failure cases.

Chapter 5

Bootstrapping the Performance of Webly Supervised Segmentation

5.1 Introduction

In the last chapter, we have addressed the problem of weakly supervised semantic segmentation by proposing a framework based on web image co-segmentation. The main idea of the framework was to train a "mask generator" using cheap data from the internet and use it to generate pixel-level masks for the training images of the target dataset. In this chapter, we follow the similar idea and propose a new framework for generating high-quality masks, closer in accuracy to those created by humans. This new framework does not rely on co-segmentation methods and gives better estimation of the masks.

The new framework is similar to the previous method in another aspect that they are all webly supervised methods. Web data exist in large quantities and we can easily collect a group of images associated with a particular class label by using the label (and synonyms) as a query to a search engine. The hope is that these extra data can be used to boost the performance, and indeed a number of papers (Wei et al. [2017b]; Jin et al. [2017]; Shen et al. [2017a]) have previously explored this idea to improve results of weakly supervised methods. There are two hurdles to overcome; the first is that the retrieved web data will often be noisy, in the sense that the image labels (tags) may not match the image content, or be inconsistent with the

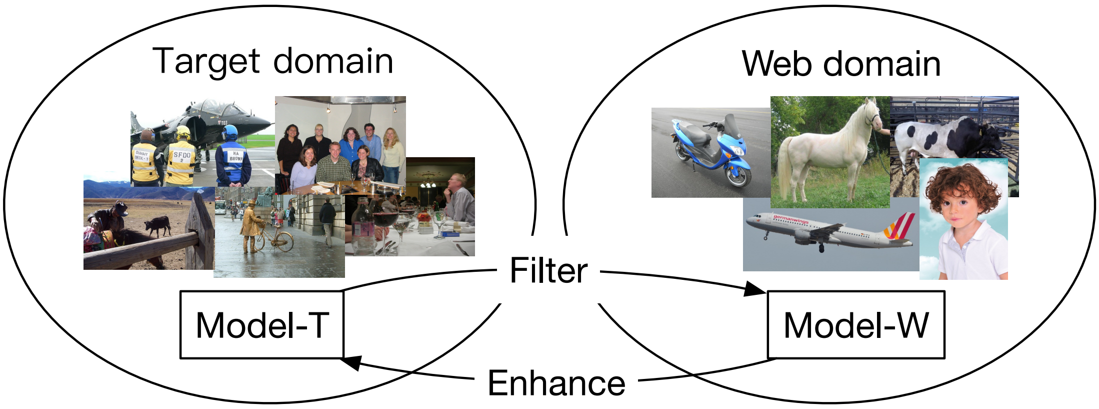


Figure 5.1: Illustration of the bi-directional framework. Model-T and Model-W are trained in the target domain and the web domain respectively. Model-T uses the knowledge in its domain to help Model-W to filter out image with incorrect tags, yielding a set of high quality easy images. Model-W trained with high quality web images transfers the knowledge back to the target domain, helping Model-T enhance the results.

concept/object we are trying to capture. The second is of course that the retrieved images will not have the ground-truth segmentation masks associated with them.

In this chapter, we describe a bootstrapping process, in which we leverage bi-directional flow of information between two domains, a target domain (i.e. the set of classes for which we want segmentation and a set of training images with accurate image-level labels) and the web domain (i.e. images crawled from the web using the target class labels as search keywords). For simplicity, we use Model-T and Model-W to represent models in the target and web domain respectively (see Figure 5.1). The key insight is that we can use a weakly supervised network (Model-T, trained on the target domain using only image-level labels) to effectively filter the web-retrieved images to eliminate labelling errors and to retain only images that are relatively easy to segment, having a simple background, single semantic class, and decent-sized objects. By doing this, we create a new dataset with high quality images that are easier to segment with only weak supervision. Figure 5.2 illustrates typical images and segmentation results from the two domains. Images in the target domain usually have a complex scene and multiple, overlapping objects, whereas web images filtered are simpler and therefore easier to segment using a weakly supervised network.

Since the model trained with the target dataset can filter the web data and provides us with a high quality dataset, we propose to learn a model with these web images and in return help enhance our results. As shown in Figure 5.3, the first two masks are estimated by the model trained with the target dataset and web images respectively. We observe that the model trained with the target dataset is good at distinguishing semantic classes but provides bad boundaries, while the model trained with web images gives good boundaries but tends to merge different semantic regions. By our merging strategy, the enhanced mask, shown in right bottom of Figure 5.3, takes advantage from both masks and makes high quality estimation. There is also the ground truth annotation in upper right for visual comparison, which is not available in our weakly supervised setting.

Our contributions can be summarized as follows:

- We propose a bidirectional transfer learning framework for bootstrapping webly supervised semantic segmentation.
- We propose an effective approach to filter web data and find high quality images, which are suitable for weakly supervised semantic segmentation.
- We transfer the knowledge learnt from the web domain to the target domain and generate high quality masks.
- By using the high quality masks as proxy ground truth, we train a standard FCN and achieve state-of-the-art performance. The gap between weakly supervised methods and fully supervised methods is further reduced.

5.2 Background

Semantic segmentation has greatly benefited from FCN based networks that enable training dense prediction models in an end-to-end fashion. Many methods have been

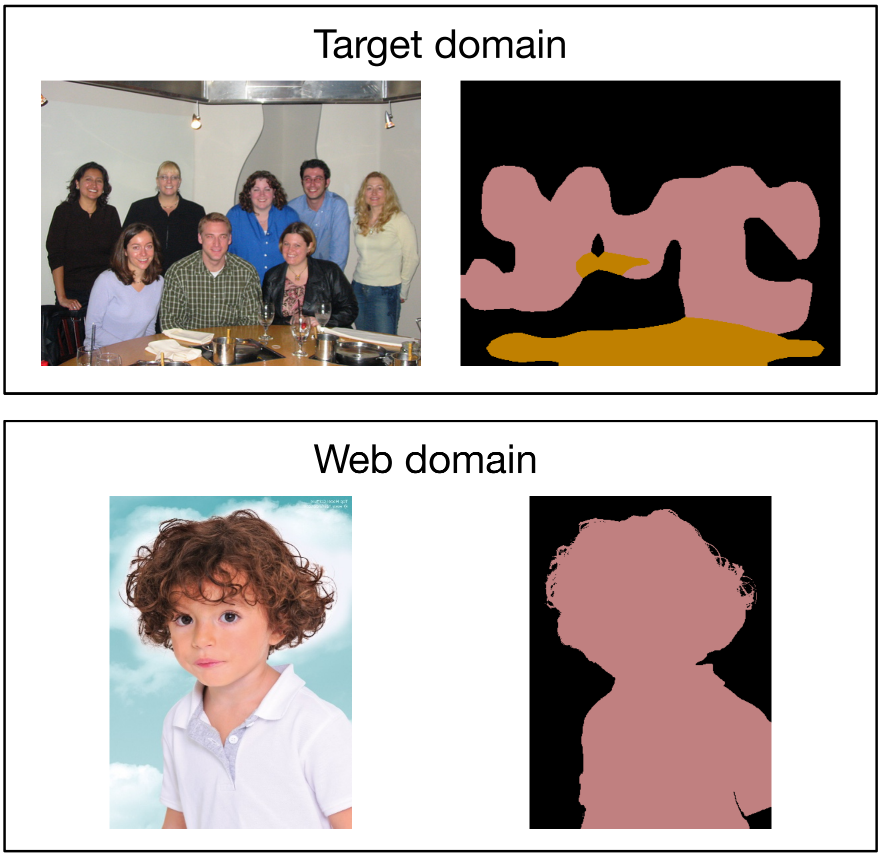


Figure 5.2: Mask estimation in two domains. In the upper part, the mask is given by the model trained in the target domain, which is coarse due to complex scene and overlapping objects of the images. The lower part shows an example given by the model trained in the web domain, which is better because of the simple context.

proposed (Long et al. [2015]; Lin et al. [2016b]; Chen et al. [2015a]; Lin et al. [2017a]; Noh et al. [2015]; Zhao et al. [2017]) and achieved remarkable success. However these methods are designed in fully supervised setting and require pixel-level masks, which involves a large amount of human labour and time to obtain.

In order to reduce the effort of annotation, many semi- and weakly supervised methods have been proposed (Pinheiro and Collobert [2015]; Bearman et al. [2016]; Shen et al. [2017a]; Lin et al. [2016a]; Kolesnikov and Lampert [2016]; Wei et al. [2017b]; Dai et al. [2015]; Pathak et al. [2015]). In these methods, various forms of supervision are investigated to achieve reasonable performance compared with fully supervised methods. Dai et al. [2015] propose a bounding box supervised

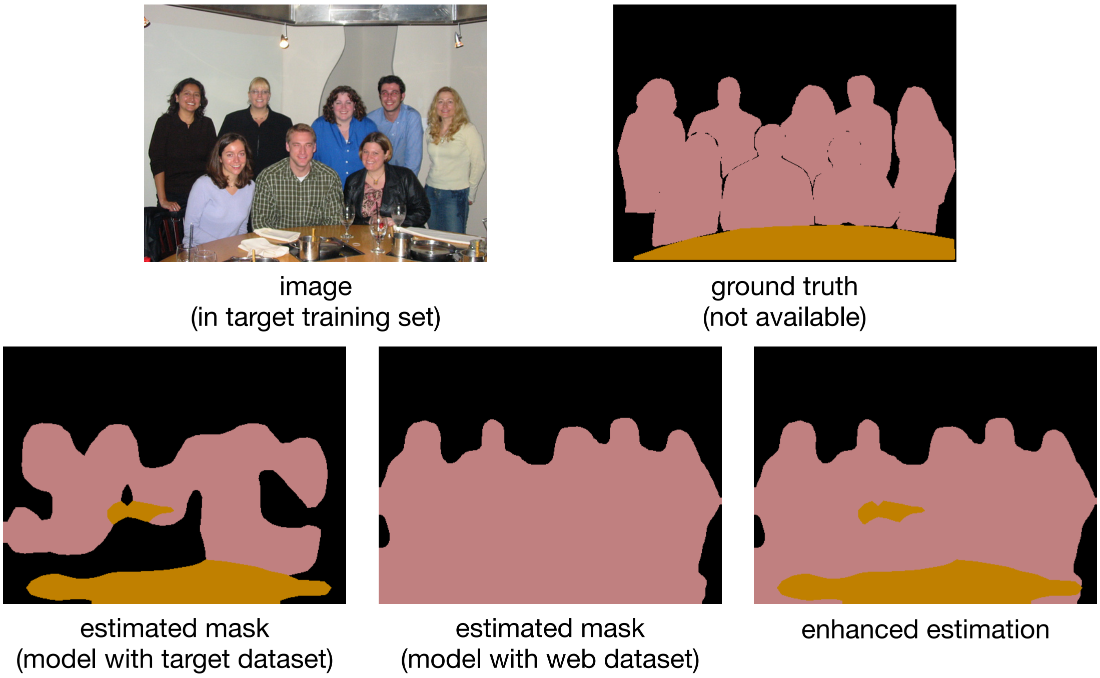


Figure 5.3: Illustration of enhancing mask. The upper part shows an image in the training set and ground truth (which is not available in our weakly supervised setting). In the lower part, the first two masks are estimated by the model in the target domain and the web domain respectively and the last one is the enhanced mask.

method where they extract object masks based on the bounding box by using MCG. Lin et al. [2016a] use scribbles as supervision and construct a graphical model to tackle the problem. In Bearman et al. [2016], only points are used as supervision to train a model. Among these supervisions, the most challenging one is image-level annotation. Pathak et al. [2015] introduce a constrained convolutional neural network with assumptions on object size, foreground and background. Pinheiro and Collobert [2015] propose a Multiple Instance Learning (MIL) based method for the problem. In Kolesnikov and Lampert [2016], a "seed, expand and constrain" (SEC) framework is proposed using only image-level labels where localization cues from classification networks are used to find the object; a weighted rank pooling loss is used to constrain the object extent; CRF is used to refine the boundaries. Our method uses SEC model as a starting point and use web images to learn better features.

Our method is closely related to webly supervised learning (Chen and Gupta [2015]; Krause et al. [2016]; Xiao et al. [2015]; Chen et al. [2013]), which is focused on extracting useful knowledge or features from noisy web data. Many webly based semantic segmentation methods have also been proposed (Jin et al. [2017]; Shen et al. [2017a]; Wei et al. [2017b]; Hong et al. [2017]). In Jin et al. [2017]; Wei et al. [2017b], a network is firstly trained with simple images from the internet and the corresponding masks estimated using saliency detection. Then the network is adapted to the target domain with progressive improvement. Shen et al. [2017a] use co-segmentation to extract the masks of web images and train the network. Hong et al. [2017] use data from the web crawled videos and extract masks based on temporal information and attention cues.

5.3 Method

The pipeline of our framework is described in Figure 5.4. Our goal is to estimate the masks for training images in the target domain, which will then be used as a proxy for ground truth to train the final segmentation network. The models in two domains interact with each other to transfer knowledge and finally provide us with high quality masks for the training images.

In detail, our bi-directional framework is based on the two domains:

- In the **target domain**, we train Initial-SEC on VOC images with only image-level labels and get initial estimation of the masks. Details are presented in Section 5.3.1.

- In the **web domain**, we transfer the knowledge from target domain by using Initial-SEC as a filter to clean noisy web data. Then we have three steps to learn the knowledge from the web domain by training Web-SEC (Section 5.3.2.2), using Grabcut refinement (Section 5.3.2.3) and training Web-FCN (Section 5.3.2.4).

- Back to the **target domain**, we transfer the knowledge from the web domain

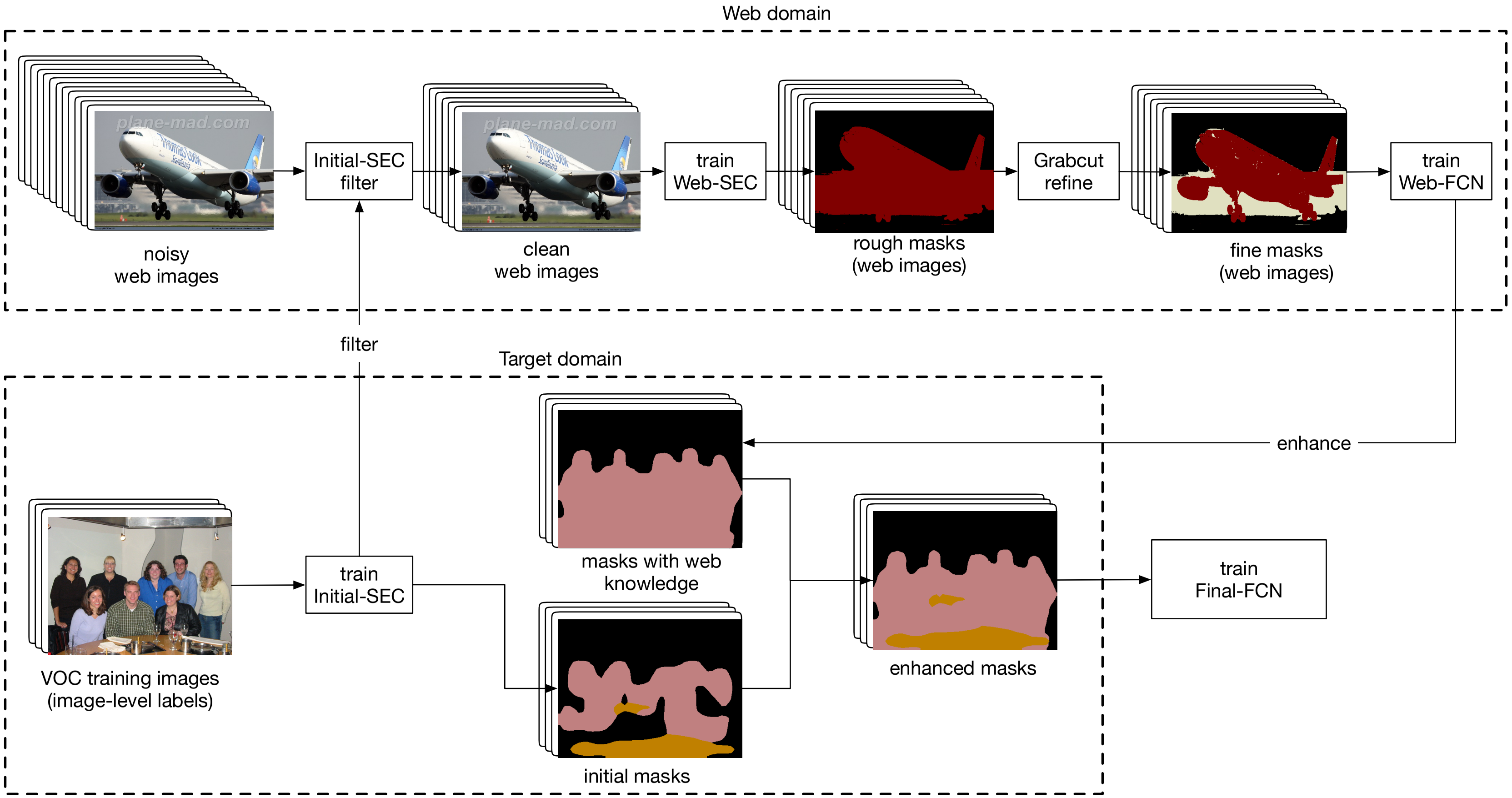


Figure 5.4: Illustration of our pipeline. Assuming the target dataset is PASCAL VOC 2012, the target domain contains the training images in VOC with image-level labels, shown in the lower rectangle with dashed lines. The web domain has noisy (i.e. incorrectly labelled) images, represented in the upper rectangle with dashed lines. Beginning with the target domain, we first train Initial-SEC to generate rough initial masks. We then use this model as a filter to clean the noisy web data and remove complex images, retaining easy-to-segment ones. In the (filtered) web domain, we train another SEC model (Web-SEC) to get rough masks for the web images and Grabcut refinement to further refine the masks. Then a FCN (Web-FCN) is trained on these data to represent the knowledge in the web domain. This model in turn enhances the estimation of the initial masks to generate high quality masks. The last step is to train Final-FCN using the proxy ground truth.

back to enhance the initial estimation of the masks, which is described in Section 5.3.3.

- Finally Final-FCN is trained using the estimated masks, as described in Section 5.3.4.

5.3.1 Training Initial-SEC in the Target Domain

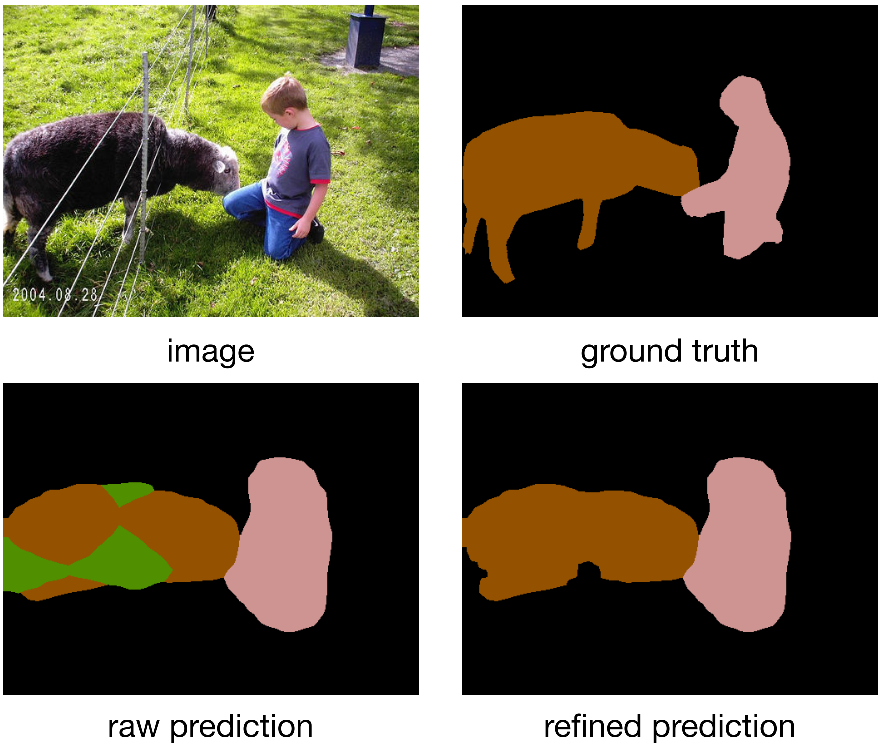
Our framework starts in the target domain, where we train a SEC model, termed Initial-SEC, on VOC images. We first review the SEC architecture Kolesnikov and Lampert [2016]. Let $I = \{(\mathbf{X}_n, \mathbf{Y}_n)\}^{N_1}$ be our target dataset, e.g. PASCAL VOC 2012, which consists of N_1 images. Each Image $\mathbf{X}_n$ is annotated by image-level labels $\mathbf{Y}_n \in \{0, 1\}^C$ where C is the number of classes. The goal is to train a DCNN $f(\mathbf{X})$, short for $f(\mathbf{X};`)$, that is parameterized by ` and models category probabilities for each pixel. The SEC model is trained by three losses:

$$\begin{aligned} \mathcal{L} = \sum_{n}^{N_1} \mathcal{L}_{seed}(f(\mathbf{X}_n), \mathbf{Y}_n) + \mathcal{L}_{expand}(f(\mathbf{X}_n), \mathbf{Y}_n) \\ + \mathcal{L}_{constrain}(f(\mathbf{X}_n), \mathbf{X}_n) \end{aligned} \tag{5.1}$$

$\mathcal{L}_{seed}$ supervises the network with localization cues obtained from Class Activation Mapping (CAM) Zhou et al. [2016a]. $\mathcal{L}_{expand}$ controls how to aggregate the heat maps to be consistent with image-level labels where a global weighted rank pooling (GWRP) is proposed. $\mathcal{L}_{constrain}$ makes the predictions respect the boundaries of objects.

In the original paper, the trained model is the final model. Unlike their approach, we apply the model back to the training images to generate their masks. These masks are coarse, as shown in left bottom of Figure 5.3, and will be enhanced by the model trained in the web domain.

Since we have access to image-level labels, we use them to further refine the masks



image

ground truth

raw prediction

refined prediction

Figure 5.5: Illustration of removing confusions of the initial masks by using image-level labels. Given an image in upper left, the raw estimation is shown in lower left. Using this information, we get cleaned estimation in right bottom. We also use the ground truth annotation for visual comparison(not available in our setting).

of the training images as follows:

$$m_i = \operatorname*{arg\,max}_{j \in \{1,..,C\}} y_i f_{ij} \tag{5.2}$$

where m_i is the mask prediction for ith pixel (i.e. we choose the class label as the most likely one from the set of valid labels). An example is illustrated in Figure 5.5. Compared with the raw prediction on left bottom, the confusion is removed in the refined prediction shown on right bottom. We also use the ground truth annotation for visual comparison, which is not available in our setting.

5.3.2 Training Models in the Web Domain

The masks estimated from Section 5.3.1 are still too rough to be used as the ground truth, as shown in right upper of Figure 5.2. In this section we show how we can leverage web-crawled data, transferring knowledge from the target domain to the web domain and learn new knowledge in the web domain.

5.3.2.1 Crawl and Filter Web Images

High quality web data processed by good filtering methods are crucial to learning good segmentation models. In this section, we show how to transfer the knowledge from the target domain to filter web data.

We first search for images based on class names using search engines (Bing in our experiments). The class names are used as seeds, along with synonyms, and similar words suggested by the search engines. For example, when searching for "dog", "German Shepherd dog", "Pitbull dog" etc. are also suggested. After greedily crawling all related images, we use the Initial-SEC model trained on VOC images as our filter to clean the web data.

Applying the SEC model to web images, we are able to obtain masks with per-pixel class labels. Based on the dense masks information, we can easily identify qualified images by scene complexity of the image, extent of the object and semantic relevance. Specifically, we select the images according to two criteria: (i) the number of pixels for the target class must lie in a predefined range, $t_1 < \frac{1}{N}\sum_i \mathbb{1}(m_i = c) < t_2$; and (ii) the number of other foreground pixels should be lower than a threshold, $\frac{1}{N}\sum_i \mathbb{1}(m_i \neq c \,\text{and}\, m_i \neq \textit{background}) < t_3$. The intuition is we want to select images with a "proper" size for the foreground. It is expected that such images can be easily segmented. Different from existing filtering approaches Wei et al. [2017b]; Hong et al. [2017], our method is based on dense masks and provides richer information of the images.

5.3.2.2 Training of Web-SEC

The filtering process described above creates a dataset of accurately labelled, high quality web images from a noisy web search. Our goal now is to improve the estimates of their masks. To this end, we train another SEC model on the web data which we term "Web-SEC". Unlike in the target domain, where images are associated with multiple class labels, images in the web domain are much simpler, filtered to be likely to contain only one class, and therefore easier to segment.

The Web-SEC model is able to generate masks for these web images of higher quality than Initial-SEC. Figure 5.6 shows a qualitative comparison between these two models (Initial-SEC and Web-SEC). The middle masks are from Initial-SEC trained in the target domain. It gives basic semantic information and rough extent of the object. Clearly the masks on the right, outputs from Web-SEC, are well adapted to the web domain and provide more accurate estimation.

5.3.2.3 Grabcut Refinement

The masks generated by Web-SEC are good at capturing the whole object but sometimes overestimate the object, as illustrated in the second column of Figure 5.7. To further refine the masks, we develop a Grabcut based refinement method. It is similar to Khoreva et al. [2017], but we use the mask as prior knowledge to indicate the foreground and background instead of the bound box. We simply jitter the window that tightly surrounds the mask and perform Grabcut Rother et al. [2004]. By multiple samples, we are able to get a probability heat map of the foreground as shown in the third column of Figure 5.7, and we retain as foreground only the pixels with high probability.

For a mask estimated by Web-SEC, $l_i \in \{1, ..., C\}$ is the label for ith pixel. After Grabcut refinement, we have $p_i \in [0, 1]$ for ith pixel representing the probability of

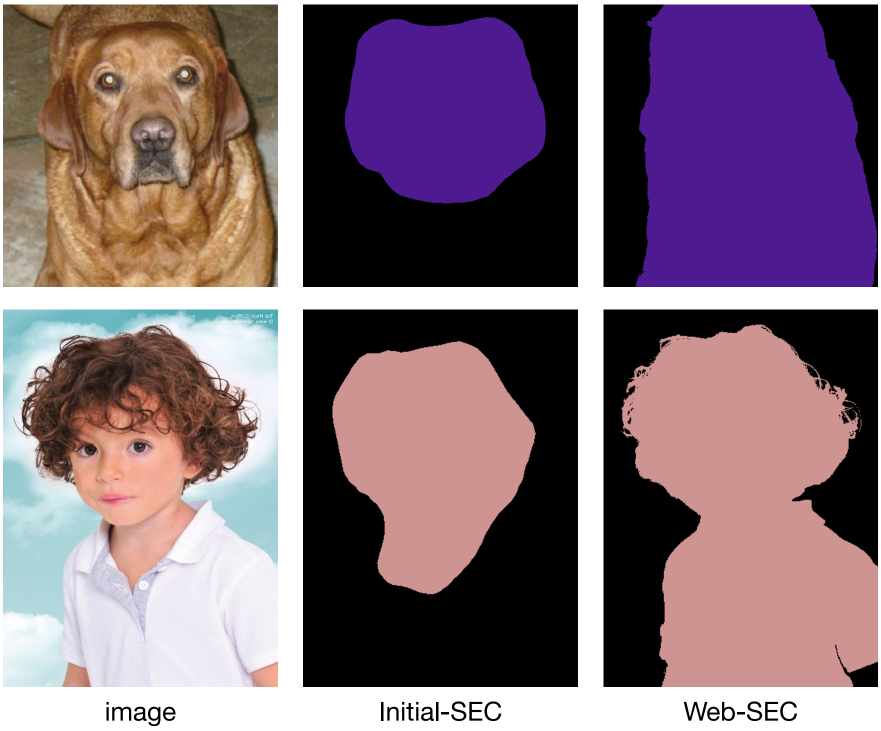


Figure 5.6: Comparison of the estimated mask for web images between Initial-SEC model and Web-SEC model. The middle column shows the masks estimated from Initial-SEC model, which are coarse. The masks on the right are from the Web-SEC model, which provide more accurate estimation.

being kept. The refined mask is defined as:

$$\hat{l}_i = \begin{cases} l_i & \text{if } p_i \geq t \\ \mathit{background} & \text{if } p_i < t \text{ and } l_i = \mathit{background} \\ \mathit{void} & \text{if } p_i < t \text{ and } l_i \neq \mathit{background} \end{cases} \tag{5.3}$$

where $\hat{l}_i$ is the new label for ith pixel; t is the threshold; *void* indicates unclear regions.

We are able to control the balance between precision and recall by choosing a proper threshold. By using a high threshold, we have high confidence about the pixels being kept. Since those with low probability are set to void, they will be ignored during the training and not have a big impact.

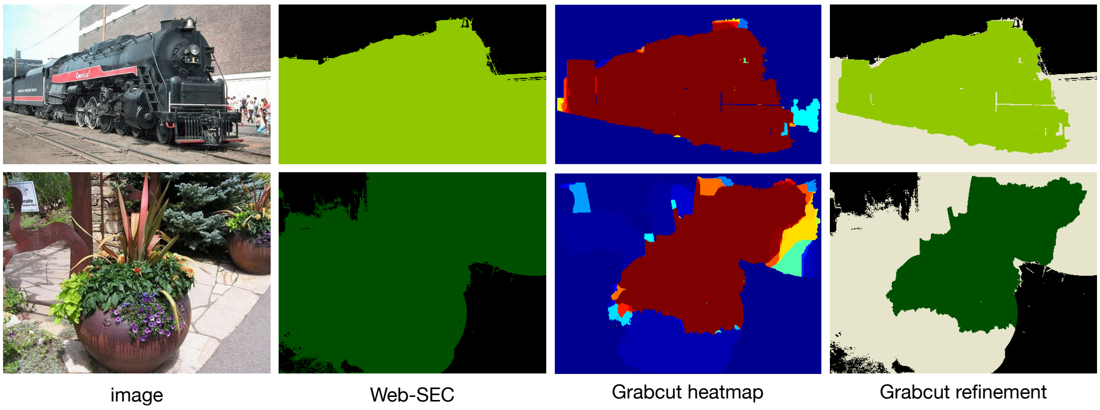


Figure 5.7: Illustration of Grabcut refinement. The second column shows the masks from Web-SEC model. The third column shows the probability heat map after Grabcut. The last column shows the refined masks.

5.3.2.4 Training of Web-FCN

After Section 5.3.2.3, we obtain a web image dataset with estimated masks. Let $W = \{(\mathbf{X}_n, \mathbf{M}_n)\}^{N_2}$ be the dataset with N_2 images, where $\mathbf{X}_n$ and $\mathbf{M}_n$ are the image and the estimated mask respectively. We now are able to train a standard FCN (Web-FCN), which is used to estimate masks for our target dataset. The architecture we adopt here is a 1/8 resolution FCN with dilated convolution kernels, similar to DeepLab Chen et al. [2015a]. This becomes a "fully supervised" problem and the objective is to minimize a softmax loss:

$$\mathcal{L} = \sum_{n}^{N_2} \mathcal{L}_{softmax}(f(\mathbf{X}_n), \mathbf{M}_n) \tag{5.4}$$

The Web-FCN trained in the web domain encodes the knowledge in this domain. The knowledge will be transferred to the target domain by applying this model to the target dataset.

5.3.3 Enhancing the Initial Estimation

In this section, we describe how to transfer the knowledge learnt from the web domain to the target domain and improve the estimation. Recall that in lower part of

Figure 5.3, the first two masks are from models in the target and the web domain respectively. We observe that the model in the target domain is good at distinguishing classes because it is trained with confident image-level labels. In contrast, the model in the web domain provides better boundaries and captures more complete extent but is prone to making mistakes about the class labels. We address this by fusing the estimations from both domains and get the final enhanced mask, shown in right bottom of Figure 5.3.

More specifically, let $M^{(t)}$ be the mask from the target domain and $M_i^{(t)} \in \{1, ..., C\}$ represent the category for ith pixel. Likewise, $M^{(w)}$ and $M^{(f)}$ represent the mask from the web domain and the final enhanced mask respectively. The fusion strategy is as follows:

$$M_i^{(f)} = \begin{cases} M_i^{(t)} & \text{if } M_i^{(w)} \neq \mathit{background} \\ M_i^{(t)} & \text{if } M_i^{(w)} = \mathit{background} \\ & \text{and } \sum_k \mathbb{1}(M_k^{(w)} = M_i^{(t)}) < \epsilon \\ M_i^{(w)} & \text{otherwise} \end{cases} \tag{5.5}$$

where ϵ is a small number.

The intuition for this strategy is that for foreground pixels in $M^{(w)}$, the category labels will follow $M^{(t)}$ because it has better ability to distinguish classes. For background pixels in $M^{(w)}$, if the number of pixels for a valid class is lower than a threshold, we also follow the label in $M^{(t)}$. This indicates if a class is shown in image-level labels, we should guarantee some pixels for this class, otherwise the information for this class will be lost. In any other cases, we follow $M^{(w)}$.

5.3.4 Training Final-FCN

After obtaining the enhanced masks, the problem is similar to a "fully supervised" problem. The target dataset becomes $I = \{(\mathbf{X}_n, \mathbf{Y}_n, \mathbf{M}_n)\}^{N_1}$, where we have pixel-wise masks besides image-level labels. This enables us to train a standard FCN model.

The structure we adopt in our experiment is a FCN with dilated kernels, similar to DeepLab Chen et al. [2015a]. Besides, we also adopt a global-multi label branch for scene consistency, as in Shen et al. [2017b]. We train Final-FCN by minimizing two loss functions:

$$\mathcal{L} = \sum_{n}^{N_1} \mathcal{L}_{softmax}(f(\mathbf{X}_n), \mathbf{M}_n) + \mathcal{L}_{multi}(g(\mathbf{X}_n), \mathbf{Y}_n) \tag{5.6}$$

where $g(\mathbf{X}_n)$ is the output for global multi-label and $\mathcal{L}_{multi}$ is a logistic multi-label loss.

5.4 Experiments

5.4.1 Dataset

Retrieved Dataset: We retrieve images from Bing based on class names. We use class names as seeds and greedily search for related images, including synonyms, words suggested by the searching engine. By using our Initial-SEC as a filter and setting a threshold for each class as the maximum number of images, we obtain a retrieved dataset with 76683 images. All images are resized so that the larger dimension is 500. In term of the parameters mentioned in Section 5.3.2.1, $t_1 = 0.3$, $t_2 = 0.7$ and $t_3 = 0.1$.

PASCAL VOC 2012: We use this dataset as our target dataset and evaluate the performance based on this. The original dataset Everingham et al. [2010] contains 1464 training images, 1449 validation images and 1456 testing images. As common practice, we also use the augmented data from Hariharan et al. [2011], which gives 10582 training images in total. There are 21 classes including a background class. The result is evaluated with Intersection over Union (IoU) averaged over 21 classes.

5.4.2 Implementation Details

The implementation is based on MXNet Chen et al. [2015b]. For details of training SEC models, Initial-SEC and Web-SEC, please refer to the original paper Kolesnikov and Lampert [2016]. We follow the same parameters except that for training Web-SEC, we use a smaller initial learning rate of 1e-4. For Grabcut refinement, Section 5.3.2.3, we set the threshold $t = 0.7$. For Web-FCN we use DeepLab-based Chen et al. [2015a] structure, which has output resolution of 1/8. For Final-FCN, apart from the basic structure, a global multi-label branch is also introduced to encourage scene consistency, similar to Shen et al. [2017b]. We use standard Stochastic Gradient Descent (SGD) for optimization. For post-processing, multi-scale inference and dense-CRF are used as common practice.

5.4.3 Experiment Results

The results on PASCAL VOC validation set and test set are shown in Table 5.4 and Table 5.5 respectively. According to the tables, the one with VGG16 Simonyan and Zisserman [2015], same as the other's base network, already achieves state of the art performance, 60.2[1]. By using another base net, Resnet 50 He et al. [2016], we achieve much better result 63.9[2], which significantly outperforms other methods.

Table 5.1 also shows a comparison with methods using different supervision, where the extra supervision is explained in the last column. In the upper half of the table, we list methods with stronger supervision than image-level labels. It is worth noting that our method does not use any other auxiliary methods that involve extra supervision. Some qualitative examples are shown in Figure 5.8.

[1] http://host.robots.ox.ac.uk:8080/anonymous/X0CH0F.html

[2] http://host.robots.ox.ac.uk:8080/anonymous/GKJXB6.html

Method	val	test	Extra Supervision
Chen Chen et al. [2015a]	**63.7**	**66.4**	Fully supervised
Lin Lin et al. [2016a]	63.1	-	Scribble
Dai Dai et al. [2015]	62.0	64.6	Bounding box+MCG
Oh Oh et al. [2017]	55.7	56.7	Bounding box
Bearman Bearman et al. [2016]	46.1	-	Point
Wei Wei et al. [2017a]	55.0	55.7	Supervised saliency
STC Wei et al. [2017b]	49.8	51.2	Supervised saliency
EM-Adapt Papandreou et al. [2015]	33.8	39.6	-
CCNN Pathak et al. [2015]	35.3	35.6	-
SEC Kolesnikov and Lampert [2016]	50.7	51.7	-
Hong Hong et al. [2017]	58.1	58.7	-
Ours-VGG16	58.8	60.2	-
Ours-Res50	**63.0**	**63.9**	-

Table 5.1: Comparison with methods using other supervisions.

T-domain	Web-domain				
Initial-SEC	Web-SEC	GC	Web-FCN	post	IoU
✓					49.3
✓	✓				52.6
✓	✓		✓		55.7
✓	✓	✓	✓		56.6
✓	✓	✓	✓	✓	**58.8**

Table 5.2: Comparison under different settings on the PASCAL VOC 2012 validation set.

Number of web images	IoU
76.7k	**56.6**
58.1k	56.4
39.1k	56.3
20.0k	56.4
10.0k	56.4
6k	55.7
2k	55.3
80.0k without filtering	49.8

Table 5.3: Ablation study using different number of web images on the PASCAL VOC 2012 validation set.

Method	bk	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mean
EM-Adapt	67.2	29.2	17.6	28.6	22.2	29.6	47.0	44.0	44.2	14.6	35.1	24.9	41.0	34.8	41.6	32.1	24.8	37.4	24.0	38.1	31.6	33.8
CCNN	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
MIL+seg	79.6	50.2	21.6	40.9	34.9	40.5	45.9	51.5	60.6	12.6	51.2	11.6	56.8	52.9	44.8	42.7	31.2	55.4	21.5	38.8	36.9	42.0
SEC	82.4	62.9	26.4	61.6	27.6	38.1	66.6	62.7	75.2	22.1	53.5	28.3	65.8	57.8	62.3	52.5	32.5	62.6	32.1	45.4	45.3	50.7
STC	84.5	68.0	19.5	60.5	42.5	44.8	68.4	64.0	64.8	14.5	52.0	22.8	58.0	55.3	57.8	60.5	40.6	56.7	23.0	57.1	31.2	49.8
WebS	84.3	65.3	27.4	65.4	**53.9**	46.3	70.1	69.8	79.4	13.8	61.1	17.4	73.8	58.1	57.8	56.2	35.7	66.5	22.0	50.1	46.2	53.4
CrawlSeg	**87.0**	69.3	32.2	70.2	31.2	58.4	73.6	68.5	76.5	26.8	63.8	29.1	73.5	69.5	66.5	70.4	46.8	72.1	27.3	**57.4**	50.2	58.1
Ours-VGG16	85.0	**74.4**	24.9	76.2	20.7	58.2	82.3	**73.6**	81.0	25.9	71.3	37.4	71.8	69.6	70.3	71.0	44.1	73.8	34.1	48.4	40.0	58.8
Ours-Resnet50	86.8	71.2	**32.4**	**77.0**	24.4	**69.8**	**85.3**	71.9	**86.5**	**27.6**	**78.9**	**40.7**	**78.5**	**79.1**	**72.7**	**73.1**	**49.6**	**74.8**	**36.1**	48.1	**59.2**	**63.0**

Table 5.4: Results on the PASCAL VOC 2012 validation set, compared with other methods, EM-Adapt (Papandreou et al. [2015]), CCNN (Pathak et al. [2015]), MIL+seg (Pinheiro and Collobert [2015]), SEC (Kolesnikov and Lampert [2016]), STC (Wei et al. [2017b]), WebS (Jin et al. [2017]) and CrawlSeg (Hong et al. [2017]).

Method	bk	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mean
EM-Adapt	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
CCNN	70.1	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+seg	78.7	48.0	21.2	31.1	28.4	35.1	51.4	55.5	52.8	7.8	56.2	19.9	53.8	50.3	40.0	38.6	27.8	51.8	24.7	33.3	46.3	40.6
SEC	83.5	56.4	28.5	64.1	23.6	46.5	70.6	58.5	71.3	23.2	54.0	28.0	68.1	62.1	70.0	55.0	38.4	58.0	39.9	38.4	48.3	51.7
STC	85.2	62.7	21.1	58.0	31.4	55.0	68.8	63.9	63.7	14.2	57.6	28.3	63.0	59.8	67.6	61.7	42.9	61.0	23.2	52.4	33.1	51.2
WebS	85.8	66.1	30.0	64.1	**47.9**	58.6	70.7	68.5	75.2	11.3	62.6	19.0	75.6	67.2	72.8	61.4	44.7	71.5	23.1	42.3	43.6	55.3
Hong	**87.2**	63.9	**32.8**	72.4	26.7	64.0	72.1	70.5	77.8	23.9	63.6	32.1	77.2	75.3	76.2	71.5	45.0	68.8	35.5	**46.2**	49.3	58.7
Ours-VGG16	85.3	**77.6**	26.2	**76.6**	17.3	61.4	82.4	74.8	83.8	25.7	66.9	46.2	74.0	75.6	79.2	70.8	48.3	73.1	40.5	38.8	39.0	60.2
Ours-Resnet50	**87.2**	76.8	31.6	72.9	19.1	**64.9**	**86.7**	**75.4**	**86.8**	**30.0**	**76.6**	**48.5**	**80.5**	**79.9**	**79.7**	**72.6**	**50.1**	**83.5**	**48.3**	39.6	**52.2**	**63.9**

Table 5.5: Results on the PASCAL VOC 2012 test set, compared with other methods, EM-Adapt (Papandreou et al. [2015]), CCNN (Pathak et al. [2015]), MIL+seg (Pinheiro and Collobert [2015]), SEC (Kolesnikov and Lampert [2016]), STC (Wei et al. [2017b]), WebS (Jin et al. [2017]) and CrawlSeg (Hong et al. [2017]).

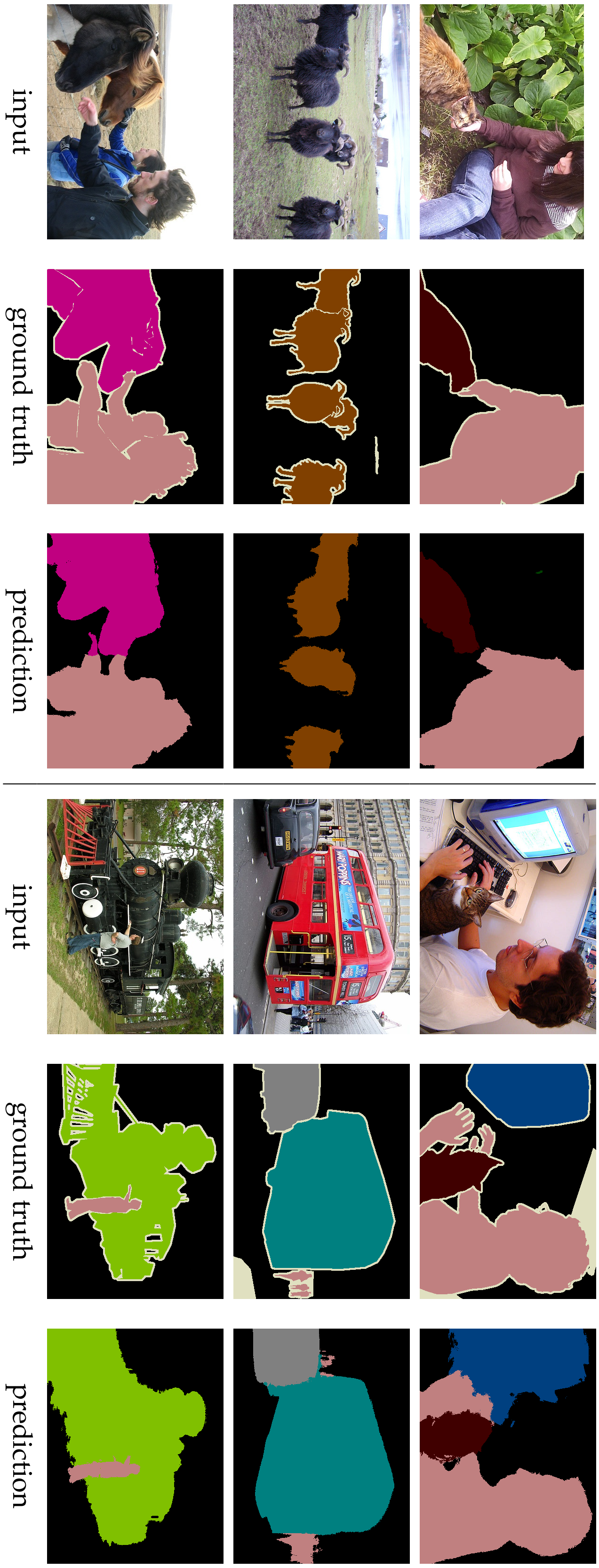


Figure 5.8: Qualitative results on PASCAL VOC 2012 validation set.

5.4.4 Ablation Study

5.4.4.1 Analysis of Different Modules

To analyse the effectiveness of our bi-directional transfer learning framework, we conduct ablation study with different settings. Recall that our goal is to generate high quality masks for the training images and train a FCN using the estimated masks. Therefore, the quality of the masks directly affects the final performance. Table 5.2 shows a comparison under different settings. Starting with the simplest one where only target domain is involved, we only get 49.3 by using Initial-SEC. With the web domain introduced, we train Web-SEC for the web images, which gives us 3.3 point improvement. This indicates the effectiveness of the knowledge transferred from the web domain. We continue training Web-FCN without using Grabcut refinement and further improve the result to 55.7. By using Grabcut refinements, we get almost one more point of improvement. The final score is obtained by post-processing including multi-scale inference and dense-CRF as common practice.

5.4.4.2 Analysis of Number of Web Images

It is also interesting to analyse how the number of web images involved affects the result. Table 5.3 shows an ablation study using different numbers of web images.The best performance is obtained by using 76.7k images. We also run experiments with different numbers of images by varying the threshold of maximum images for each class. It is interesting that the performance does not drop much with the number of web images decreasing. Even the number of images is decreased to 2k, the performance only drops by 1.3%. This indicates that our bi-directional framework is pretty robust to noise and the filtered images are high quality. Furthermore, we also show an experiment without filtering the images, which is shown in the last row. Using 80k noisy web images, we only get score of 49.8, which is 6.8 lower than the best one. This again indicates the importance of using knowledge learnt in target domain to filter web data.

5.5 Conclusion

In this chapter, we tackle the problem of weakly supervised semantic segmentation using only image-level labels. Apart from the target dataset with confident image-level labels, we propose to use noisy web data to boost the performance. To leverage the data in two domains, target domain and web domain, we propose a novel bi-directional transfer learning framework that is able to generate high quality masks for the training images. Using these masks as proxy ground truth, we achieve state-of-the-art performance and further narrow down the gap between weakly and fully supervised methods.

Chapter 6

Conclusion and Future Directions

6.1 Conclusion

In this thesis, we have discussed two problems in semantic segmentation, scene consistency and weakly supervised training.

In Chapter 3, we have addressed the problem of scene consistency by introducing a dense multi-label module. With comprehensive experiments, we have shown great potential of the module in terms of enforcing scene consistency in multiple levels. Our simple yet effective module can be easily integrated into other semantic segmentation systems to give more plausible and consistent predictions.

In Chapter 4 and 5, we have discussed two methods for weakly supervised semantic segmentation using only image-level labels. The two methods share the similar idea of using web data to obtain extra free supervision and further improve the performance, but they are designed from different perspectives. Chapter 4 has presented a framework that uses co-segmentation and web data to get a web dataset with pixel-level masks and use these data to further estimate the target dataset. Chapter 5 has described a bi-directional framework with two domains, the web domain and the target domain. We have discussed how the two domains interact and transfer knowledge with each other and finally boost the performance of the model in the target domain. We have also shown state-of-the-art performance achieved by our methods.

The two problems discussed in the thesis address semantic segmentation from two different angles. The multi-label module is introduced to solve the issue of scene consistency. While the multi-label module is not the only way to deal with scene consistency issue, but it reveals a fact that explicitly imposing constraint will make models more reliable than relying on the model itself to learn everything. The other problem, weakly supervised learning, addresses the problem from how to train networks using limited information. The research on weakly supervised segmentation not only provides us with the methods to train models using limited supervision, but also shows us great insight of what the minimum information the model truly needs in order to get satisfactory results.

6.2 Future Work

We have presented some solutions to the issues stated, but the models are yet perfect and some issues remain unresolved. We point out some future directions for these two problems.

The issue of scene consistency has been discussed in Chapter 3 and the dense multi-label module has been introduced to address this. However, there have been better methods for semantic segmentation proposed since then that have achieved remarkable results (such as Chen et al. [2017]). It is worth investigating the potential of our module on these better models. From another angle, it is also important to explore a more efficient structure of the module. Since the current implementation involves several branches, which might bring high overhead when applied to real-time systems.

There are also future directions for weakly supervised methods discussed in Chapter 4 and 5. The method in Chapter 4 highly relies on the performance of co-segmentation methods. Therefore, it is interesting to explore other co-segmentation methods, or design a new co-segmentation method that is more suitable for our particular setting and can be easily integrated into the whole system. For the bi-directional framework presented Chapter 5, the whole pipeline is a little bit complicated because of the multiple stages involved. There are also heuristics in the framework, such as the filter thresholds, GrabCut refinement parameters and mask merging strategies. We are particular interested to have a cleaner and more elegant framework. We have also noticed the step of GrabCut refinement is very time-consuming because we have to sample many times for each image and there is not gpu-accelerated version. It would be better either to integrate a gpu-version of GrabCut refinement unit into the system or find other alternative methods to refine the masks.

Bibliography

Alexe, B.; Deselaers, T.; and Ferrari, V., 2012. Measuring the objectness of image windows. *TPAMI*, 34, 11 (2012), 2189–2202. (cited on page 44)

Arbeláez, P.; Pont-Tuset, J.; Barron, J.; Marques, F.; and Malik, J., 2014. Multiscale combinatorial grouping. In *CVPR*. (cited on page 44)

Arnab, A.; Jayasumana, S.; Zheng, S.; and Torr, P., 2015. Higher Order Conditional Random Fields in Deep Neural Networks. *Arxiv*, (2015), 10. (cited on page 35)

B, X. Q.; Liu, Z.; Shi, J.; Zhao, H.; and Jia, J., 2016. Augmented Feedback in Semantic Segmentation Under Image Level Supervision. In *ECCV*. (cited on pages 55 and 56)

Bearman, A.; Russakovsky, O.; Ferrari, V.; and Fei-Fei, L., 2016. What's the point: Semantic segmentation with point supervision. In *ECCV*. (cited on pages 4, 16, 43, 44, 45, 56, 64, 65, and 77)

Carreira, J.; Caseiro, R.; Batista, J.; and Sminchisescu, C., 2012. Semantic Segmentation with Second-Order Pooling. In *ECCV*, vol. 7578 LNCS, 430–443. (cited on pages 2, 14, and 22)

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L., 2015a. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In *ICLR*. (cited on pages 1, 2, 3, 19, 22, 29, 64, 73, 75, 76, and 77)

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L., 2016. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *CoRR*, (2016). (cited on pages 15, 19, 22, 35, 49, and 51)

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H., 2017. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. (2017). (cited on page 84)

Chen, T.; Li, M.; Li, Y.; Lin, M.; Wang, N.; Wang, M.; Xiao, T.; Xu, B.; Zhang, C.; and Zhang, Z., 2015b. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems. *NIPS workshop*, (2015). (cited on pages 51 and 76)

Chen, X. and Gupta, A., 2015. Webly supervised learning of convolutional networks. In *ICCV*, 1431–1439. (cited on page 66)

Chen, X.; Shrivastava, A.; and Gupta, A., 2013. NEIL: Extracting Visual Knowledge from Web Data. In *ICCV*. (cited on page 66)

Chen, X.; Shrivastava, A.; and Gupta, A., 2014. Enriching Visual Knowledge Bases via Object Discovery and Segmentation. In *CVPR*. (cited on pages 44, 46, 47, 48, and 51)

Clevert, D.-A.; Unterthiner, T.; and Hochreiter, S., 2016. FAST AND ACCURATE DEEP NETWORK LEARNING BY EXPONENTIAL LINEAR UNITS (ELUS). In *ICLR*. (cited on page 12)

Cogswell, M.; Lin, X.; Purushwalkam, S.; and Batra, D., 2014. Combining the Best of Graphical Models and ConvNets for Semantic Segmentation. *arXiv*, (2014), 13. (cited on pages 2 and 22)

Dai, J.; He, K.; and Sun, J., 2015. [M] BoxSup: Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation. In *ICCV*. (cited on pages 16, 19, 22, 45, 52, 56, 64, and 77)

Dai, J.; Wu, Y. N.; Zhou, J.; and Zhu, S. C., 2013. Cosegmentation and cosketch by unsupervised learning. In *ICCV*, 1. (cited on page 46)

DALAL, N. AND TRIGGS, B., 2005. Histograms of oriented gradients for human detection. In *CVPR*. (cited on page 1)

EVERINGHAM, M.; VAN GOOL, L.; WILLIAMS, C. K. I.; WINN, J.; AND ZISSERMAN, A., 2010. The pascal visual object classes (VOC) challenge. *IJCV*, 88, 2 (2010), 303–338. (cited on pages 30, 50, and 75)

FAKTOR, A. AND IRANI, M., 2013. Co-Segmentation by Composition. In *ICCV*. (cited on page 46)

FARABET, C.; COUPRIE, C.; NAJMAN, L.; AND LECUN, Y., 2013. Learning Hierarchical Features for Scene Labeling. *TPAMI*, 35, 8 (2013), 1915–1929. (cited on pages 14 and 22)

GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; AND MALIK, J., 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*, 580–587. (cited on pages 1, 2, 14, and 22)

GONG, Y.; JIA, Y.; LEUNG, T.; TOSHEV, A.; AND IOFFE, S., 2013. Deep Convolutional Ranking for Multilabel Image Annotation. *CoRR*, (2013), 1–9. http://arxiv.org/abs/1312.4894. (cited on page 22)

GUO, Y. AND GU, S., 2011. Multi-label classification using conditional dependency networks. In *IJCAI*, 1300–1305. (cited on pages 19 and 22)

HARIHARAN, B.; ARBEL, P.; BOURDEV, L.; MAJI, S.; AND MALIK, J., 2011. Semantic Contours from Inverse Detectors. In *ICCV*. (cited on pages 51 and 75)

HARIHARAN, B.; ARBELÁEZ, P.; GIRSHICK, R.; AND MALIK, J., 2014. Simultaneous Detection and Segmentation. *ECCV*, (2014), 297–312. (cited on pages 2, 14, and 22)

HE, K.; ZHANG, X.; REN, S.; AND SUN, J., 2016. Deep Residual Learning for Image Recognition. In *CVPR*. (cited on pages 1, 10, 13, 23, 29, 43, 49, and 76)

Hochbaum, D. S. and Singh, V., 2009. An efficient algorithm for Co-segmentation. In *ICCV*. (cited on page 46)

Hong, S.; Noh, H.; and Han, B., 2015. Decoupled Deep Neural Network for Semi-supervised Semantic Segmentation. In *NIPS*. (cited on pages 22, 45, 52, and 56)

Hong, S.; Yeo, D.; Kwak, S.; Lee, H.; and Han, B., 2017. Weakly Supervised Semantic Segmentation using Web-Crawled Videos. In *CVPR*. (cited on pages xviii, 17, 66, 70, 77, 78, and 79)

Ioffe, S. and Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Arxiv*, (2015). (cited on pages 10 and 12)

Jiang, H.; Wang, J.; Yuan, Z.; Wu, Y.; Zheng, N.; and Li, S., 2013. Salient Object Detection: A Discriminative Regional Feature Integration Approach. In *CVPR*. (cited on pages 44 and 47)

Jiang, W., 2016. CNN-RNN : A Unified Framework for Multi-label Image Classification. *CVPR*, (2016). (cited on pages 19, 22, and 23)

Jin, B.; Segovia, M. V. O.; and Susstrunk, S., 2017. Webly Supervised Semantic Segmentation. *CVPR*, (2017). (cited on pages xvii, xviii, 17, 53, 54, 61, 66, 78, and 79)

Joulin, A.; Bach, F.; and Ponce, J., 2010. Discriminative clustering for image co-segmentation. *CVPR*, (2010). (cited on page 46)

Kendall, A.; Badrinarayanan, V.; and Cipolla, R., 2015. Bayesian SegNet: model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv:1511.02680v1 [cs.CV]*, (2015). (cited on page 37)

Khoreva, A.; Benenson, R.; Hosang, J.; Hein, M.; and Schiele, B., 2017. Simple Does It: Weakly Supervised Instance and Semantic Segmentation. In *CVPR*. (cited on page 71)

Kolesnikov, A. and Lampert, C. H., 2016. Seed , Expand and Constrain : Three Principles for Weakly-Supervised Image Segmentation. In *ECCV*. (cited on pages xvii, xviii, 17, 43, 45, 46, 53, 54, 56, 64, 65, 68, 76, 77, 78, and 79)

Krähenbühl, P. and Koltun, V., 2011. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *NIPS*. (cited on pages 14 and 52)

Krause, J.; Sapp, B.; Howard, A.; Zhou, H.; Toshev, A.; Duerig, T.; Philbin, J.; and Fei-Fei, L., 2016. The Unreasonable Effectiveness of Noisy Data for Fine-Grained Recognition. In *ECCV*. (cited on page 66)

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*. (cited on pages 1, 10, 11, 12, 13, and 43)

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86, 11 (1998), 2278–2323. (cited on page 10)

Lin, D.; Dai, J.; Jia, J.; He, K.; and Sun, J., 2016a. ScribbleSup: Scribble-Supervised Convolutional Networks for Semantic Segmentation. In *CVPR*. (cited on pages 16, 44, 45, 52, 56, 64, 65, and 77)

Lin, G.; Milan, A.; Shen, C.; and Reid, I., 2017a. RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation. In *CVPR*. (cited on pages 1, 2, 15, 51, and 64)

Lin, G.; Shen, C.; Hengel, A. V. D.; and Reid, I., 2017b. Exploring Context with Deep Structured models for Semantic Segmentation. *TPAMI*, PP, 99 (2017), 1–1. (cited on pages 15, 18, 35, 36, and 37)

Lin, G.; Shen, C.; van dan Hengel, A.; and Reid, I., 2016b. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*. (cited on pages 19, 22, and 64)

Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L. D.; Girshick, R. B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L., 2014. Microsoft {COCO:} Common Objects in Context. *{arXiv}:1405.0312*, (2014), 740–755. (cited on pages xvii and 35)

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; and Reed, S., 2016. SSD: Single Shot MultiBox Detector. In *ECCV*. (cited on page 1)

Liu, W.; Rabinovich, A.; and Berg, A. C., 2015. ParseNet: Looking Wider to See Better. (cited on pages 2, 3, 18, 22, and 35)

Long, J.; Shelhamer, E.; and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In *ICCV*. (cited on pages 1, 2, 14, 15, 22, 35, 36, and 64)

Lowe, D., 1999. Object recognition from local scale-invariant features. In *ICCV*. (cited on page 1)

Maas, A. L.; Hannun, A. Y.; and Ng, A. Y., 2013. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *ICML*. (cited on page 12)

Mottaghi, R.; Chen, X.; Liu, X.; Cho, N.-g.; Lee, S.-w.; Urtasun, R.; and Yuille, A., 2014. The Role of Context for Object Detection and Semantic Segmentation in the Wild. *CVPR*, (2014). (cited on page 33)

Nair, V. and Hinton, G. E., 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML*. (cited on page 12)

Noh, H.; Hong, S.; and Han, B., 2015. Learning Deconvolution Network for Semantic Segmentation. In *ICCV*. (cited on pages 2, 15, and 64)

Oh, S. J.; Benenson, R.; Khoreva, A.; Akata, Z.; Fritz, M.; and Schiele, B., 2017. Exploiting saliency for object segmentation from image level labels. In *CVPR*. (cited on pages 45, 46, 55, 56, and 77)

PAPANDREOU, G.; CHEN, L.-C.; MURPHY, K.; AND YUILLE, A. L., 2015. Weakly- and Semi-Supervised Learning of a DCNN for Semantic Image Segmentation. In *ICCV*. (cited on pages xvii, xviii, 45, 53, 54, 56, 77, 78, and 79)

PATHAK, D.; KRAHENBUHL, P.; AND DARRELL, T., 2015. Constrained Convolutional Neural Networks for Weakly Supervised Segmentation. In *ICCV*. (cited on pages xvii, xviii, 17, 43, 45, 53, 54, 56, 64, 65, 77, 78, and 79)

PENG, C.; ZHANG, X.; YU, G.; LUO, G.; AND SUN, J., 2017. Large Kernel Matters - Improve Semantic Segmentation by Global Convolutional Network. *CoRR*, (2017). (cited on page 51)

PINHEIRO, P. H. O. AND COLLOBERT, R., 2015. From Image-level to Pixel-level Labeling with Convolutional Networks. In *CVPR*. (cited on pages xvii, xviii, 16, 44, 45, 53, 54, 64, 65, 78, and 79)

REN, S.; HE, K.; GIRSHICK, R.; AND SUN, J., 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*. (cited on page 1)

ROTHER, C.; KOLMOGOROV, V.; AND BLAKE, A., 2004. "GrabCut": interactive foreground extraction using iterated graph cuts. *TOG*, , 3 (2004), 309. (cited on page 71)

RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; JAN, C. V.; KRAUSE, J.; AND MA, S., 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115, 3 (2015). (cited on page 12)

SHEN, F. AND ZENG, G., 2016. Fast Semantic Image Segmentation with High Order Context and Guided Filtering. (2016). http://arxiv.org/abs/1605.04068. (cited on page 22)

SHEN, T.; LIN, G.; LIU, L.; SHEN, C.; AND REID, I., 2017a. Weakly Supervised Semantic Segmentation Based on Web Image Co-segmentation. In *BMVC*. (cited on pages 61, 64, and 66)

Shen, T.; Lin, G.; Shen, C.; and Reid, I., 2017b. Learning Multi-level Region Consistency with Dense Multi-label Networks for Semantic Segmentation. In *IJCAI*. (cited on pages 49, 75, and 76)

Shotton, J.; Johnson, M.; and Cipolla, R., 2008. Semantic Texton Forest for Image Categorization and Segmentation. *Proceedings of the conference on Computer Vision and Pattern Recognition*, (2008), 1–8. (cited on page 14)

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 *Eccv*. (cited on page 14)

Silberman, N.; Hoiem, D.; Kohli, P.; and Fergus, R., 2012. Indoor segmentation and support inference from RGBD images. In *ECCV*. (cited on pages 34 and 35)

Simonyan, K. and Zisserman, A., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*. (cited on pages 1, 10, 11, 13, 23, 43, and 76)

Song, S.; Lichtenberg, S. P.; and Xiao, J., 2015. SUN RGB-D: A RGB-D scene understanding benchmark suite. *CVPR*, (2015), 567–576. (cited on page 35)

Srivastava, N.; Hinton, G.; and Krizhevsky, A., 2014. Dropout : A Simple Way to Prevent Neural Networks from Overfitting. 15 (2014), 1929–1958. (cited on page 12)

Szegedy, C.; Ioffe, S.; and Vanhoucke, V. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. (cited on pages 1 and 23)

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A., 2015. Going Deeper with Convolutions. In *CVPR*. (cited on pages 10 and 13)

Wang, Z.; B, H. L.; Ouyang, W.; and B, X. W., 2016. Learnable Histogram: Statistical Context Features for Deep Neural Networks. 9905 (2016), 246–262. (cited on page 18)

Wei, Y.; Feng, J.; Liang, X.; Cheng, M.-M.; Zhao, Y.; and Yan, S., 2017a. Object Region Mining with Adversarial Erasing: A Simple Classification to Semantic Segmentation Approach. In *CVPR*. (cited on page 77)

Wei, Y.; Liang, X.; Chen, Y.; Shen, X.; Cheng, M.-M.; Zhao, Y.; and Yan, S., 2017b. STC: A Simple to Complex Framework for Weakly-supervised Semantic Segmentation. *TPAMI*, (2017). (cited on pages xvii, xviii, 17, 44, 45, 46, 47, 50, 53, 54, 56, 61, 64, 66, 70, 77, 78, and 79)

Wei, Y.; Xia, W.; Huang, J.; Ni, B.; Dong, J.; Zhao, Y.; and Member, S., 2014. CNN : Single-label to Multi-label. *CoRR*, abs/1406.5 (2014). (cited on pages 22 and 23)

Wei, Y.; Xia, W.; Min Lin; Huang, J.; Ni, B.; Dong, J.; Yan, S.; and Zhao, Y., 2016. HCP: A Flexible CNN Framework for Multi-Label Image Classification. *TPAMI*, 38, 2 (2016), 1901–1907. (cited on page 19)

Wu, Z.; Shen, C.; and Hengel, A. V. D., 2016. Bridging Category-level and Instance-level Semantic Image Segmentation. (2016). (cited on page 35)

Xiao, T.; Xia, T.; Yang, Y.; Huang, C.; and Wang, X., 2015. Learning From Massive Noisy Labeled Data for Image Classification. In *CVPR*. (cited on page 66)

Xue, X.; Zhang, W.; Zhang, J.; Wu, B.; Fan, J.; and Lu, Y., 2011. Correlative multi-label multi-instance image annotation. *ICCV*, (2011), 651–658. (cited on page 22)

Yadollahpour, P.; Batra, D.; and Shakhnarovich, G., 2013. Discriminative re-ranking of diverse segmentations. *CVPR*, (2013), 1923–1930. (cited on page 22)

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J., 2017. Pyramid Scene Parsing Network. In *CVPR*. (cited on pages 2, 3, 18, and 64)

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A., 2016a. Learning Deep Features for Discriminative Localization. In *CVPR*. (cited on page 68)

Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A., 2016b. Semantic Understanding of Scenes through the ADE20K Dataset. *arXiv*, (2016). (cited on pages 31, 32, and 33)