DISSERTATION

submitted
to the
Combined Faculty for the Natural Sciences and Mathematics
of the
Heidelberg University, Germany
for the degree of
Doctor of Natural Sciences

Put forward by
M.S. Weihao Li
Born in Hebei, China
Oral examination:

Semantic-Aware Image Analysis

Advisor: Prof. Dr. Carsten Rother

To my family

Acknowledgements

I would like to thank my supervisor Prof. Dr. Carsten Rother for giving me an excellent opportunity to work under his supervision. He not only introduces and guides me into the world of computer vision and machine learning but also creates these so wonderful research environments both in Dresden and Heidelberg. Without his supervision, this thesis cannot be generated.

In the beginning stage of this journey, I worked together with Dr. Michael Ying Yang. I also would like to thank him for his feedback and the discussions.

I want to thank all members of the Computer Vision Lab Dresden and Visual Learning Lab Heidelberg, including Ullrich, Bogdan, Jakob, Stefan, Hassan, Lynton, Lisa, Siva, Omid, Eric, Siddharth, Jens, Hendrik, Titus, Radek, Alexander Krull, Alexander Kirillov, Katrin, Frank, Uwe, Anita, Dimitrij, Holger, and Oliver. In particular, I want to thank Omid, Siva, and Hassan for the kicker games.

I greatly appreciate Antje for her administrative support.

Finally, many thanks to my parents and sisters for their continued love. I especially thank my loving wife, Sally, who has always been supporting and encouraging me both in work and life.

Abstract

Extracting and utilizing high-level semantic information from images is one of the important goals of computer vision. The ultimate objective of image analysis is to be able to understand each pixel of an image with regard to high-level semantics, *e.g.* the objects, the stuff, and their spatial, functional and semantic relations. In recent years, thanks to large labeled datasets and deep learning, great progress has been made to solve image analysis problems, such as image classification, object detection, and object pose estimation. In this work, we explore several aspects of semantic-aware image analysis. First, we explore semantic segmentation of man-made scenes using fully connected conditional random fields which can model long-range connections within the image of man-made scenes and make use of contextual information of scene structures. Second, we introduce a semantic smoothing method by exploiting the semantic information to accomplish semantic structure-preserving image smoothing. Semantic segmentation has achieved significant progress recently and has been widely used in many computer vision tasks. We observe that high-level semantic image labeling information can provide a meaningful structure prior to image smoothing naturally. Third, we present a deep object co-segmentation approach for segmenting common objects of the same class within a pair of images. To address this task, we propose a CNN-based Siamese encoder-decoder architecture. The encoder extracts high-level semantic features of the foreground objects, a mutual correlation layer detects the common objects, and finally, the decoder generates the output foreground masks for each

image. Finally, we propose an approach to localize common objects from novel object categories in a set of images. We solve this problem using a new common component activation map in which we treat the class-specific activation maps as components to discover the common components in the image set. We show that our approach can generalize on novel object categories in our experiments.

Zusammenfassung

Die Extraktion und Nutzung von semantischen Informationen aus Bildern gehört zu den wichtigsten Computer-Vision-Anwendungen. Allumfassendes Ziel von Bildanalyse ist das semantische Verständnis auf Pixelebene. Dazu gehört unter anderem die Zuordnung von Pixeln zu Objekten und Flächen, sowie ihre örtlichen, funktionalen und semantischen Zusammenhänge. In den letzten Jahren konnte auf dem Feld der Bildanalyse, insbesondere bei Klassifikation, Objekterkennung und Posenschätzung, großer Fortschritt durch annotierte Datensätze und Deep Learning erzielt werden. Diese Arbeit untersucht verschiedenste Aspekte der semantischen Bildanalyse. Erstens betrachten wir die Semantische Segmentierung urbaner Szenen mittels Dense CRFs. Dense CRFs modellieren globale Zusammenhänge in Bildern unter Nutzung des Kontextes der Struktur einer Szene. Zweitens führen wir eine Methode zur semantischen strukturerhaltenden Bildglättung, unter Nutzung von Kontextinformation, ein. Semantische Segmentierung konnte seit kurzem große Fortschritte erzielen und wird in vielen Computer-Vision-Anwendungen genutzt. Wir beobachten, dass Semantik als nützliche A-Priori-Information für natürlich wirkende Bildglättung verwendet werden kann. Drittens präsentieren wir eine Methode zur Kosegmentierung von Objekten der selben Klasse in einem Paar von Bildern unter Nutzung einer Siamese Encoder-Decoder CNN-Architektur. Der Encoder extrahiert semantische Deskriptoren der Objekte im Vordergrund, ein Mutual Correlation Layer detektiert Objekte derselben Klasse. Abschließend generiert der Decoder eine Vordergrundmaske für jedes Objekt. Viertens stellen wir eine Methode zur Lokalisierung häufiger Objekte eines Bilddatensatzes aus

unbekannten Klassen vor. Zur Lösung dieses Problems führen wir eine Common Component Activation Map ein, in welcher klassenspezifische Activation Maps zur Erkennung häufiger Komponenten im Datensatz genutzt werden. Wir zeigen in Experimenten, dass dieser Ansatz auf neue Objektkategorien generalisiert.

Contents

1 Introduction **1**

1.1 Motivation 1
1.2 Challenges 4
1.3 Contributions 6
1.4 Publications 8
1.5 Outline 9

2 Semantic Segmentation of Man-made Scenes **11**

2.1 Introduction 11
2.2 Related Work 13
2.3 Method 15
2.3.1 Fully-Connected CRF 15
2.3.2 Unary Potentials 16
2.3.3 Pairwise Potentials 17
2.3.4 Inference 17
2.3.5 Learning 19
2.4 Experiments 19
2.4.1 Datasets 19
2.4.2 Results 20
2.5 Conclusion 22

3 Semantic-Aware Image Smoothing **25**

3.1 Introduction 25
3.2 Related Work 28
3.2.1 Image Smoothing 28

3.2.2 Semantic Segmentation 29
3.3 Semantic Smoothing 30
3.3.1 Data Term 30
3.3.2 Regularization Term 30
3.3.2.1 Appearance potential 31
3.3.2.2 Edge potential 31
3.3.2.3 Semantic potential 32
3.3.3 Optimization 32
3.4 Experimental Results and Applications 33
3.4.1 Datasets 33
3.4.2 Texture Removal 34
3.4.3 Applications 34
3.4.3.1 Detail Enhancement 34
3.4.3.2 Edge Detection 38
3.4.3.3 Semantic Segmentation 40
3.5 Conclusion 42

4 Common Object Segmentation **45**
4.1 Introduction 45
4.2 Related Work 48
4.2.1 Co-Segmentation without Explicit Learning 48
4.2.2 Co-Segmentation with Learning 49
4.2.3 Interactive Co-Segmentation 49
4.2.4 CNNs for Image Segmentation 50
4.3 Method 51
4.3.1 Siamese Encoder 52
4.3.2 Mutual Correlation 52
4.3.3 Siamese Decoder 53
4.3.4 Loss Function 54
4.3.5 Group Co-Segmentation 54
4.4 Experiments 54
4.4.1 Datasets 54
4.4.2 Implementation Details and Runtime 55

4.4.3 Results . . . 56
4.4.3.1 Metrics. . . . 57
4.4.3.2 PASCAL Co-Segmentation. . . . 57
4.4.3.3 MSRC. . . . 57
4.4.3.4 Internet. . . . 59
4.4.3.5 iCoseg . . . 61
4.4.4 Ablation Study . . . 63
4.5 Conclusions . . . 63

5 Common Object Localization **65**
5.1 Introduction . . . 65
5.2 Method . . . 67
5.2.1 Class Activation Map . . . 67
5.2.2 Common Component Activation Map . . . 68
5.3 Experiments . . . 69
5.3.1 Generating Boxes . . . 69
5.3.2 Evaluation Metric . . . 69
5.3.3 Dataset . . . 70
5.3.4 Results . . . 70
5.4 Conclusion . . . 70

6 Conclusions **73**
6.1 Overview . . . 73
6.2 Outlook . . . 73

Bibliography **75**

List of Abbreviations

AC	Auto-Context
ALE	Automatic Labelling Environment
CAM	Class Activation Map
CCAM	Common Component Activation Map
CNN	Convolutional Neural Network
CPN	Convolutional Patch Networks
CRF	Conditional Random Field
DOCS	Deep Object Co-Segmentation
DT	Domain Transform
FCN	Fully Convolutional Networks
GPU	Graphics Processing Unit
HOG	Histogram of Oriented Gradient
IoU	Intersection over Union
MAP	Maximum A Posterior
MC	Mutual Correlation
MPM	Maximum Posterior Marginal
RG	Region Covariance
RTV	Relative Total Variation
R-CNN	Regions with CNN features
SF	Semantic Filtering
SIFT	Scale-Invariant Feature Transform

Chapter 1

Introduction

The objective of this thesis is to deal with the problem of semantic-aware image analysis. Extracting and utilizing high-level semantic information from images is one important goal of computer vision. In recent years, large progress has been made in solving image analysis problems, such as image classification [70, 118, 121, 47], object detection [41, 40, 101, 88], and object pose estimation [11, 71, 12, 93]. In particular, thanks to large labeled datasets and deep learning, pixel-level semantic segmentation has become more and more popular [89, 17, 4, 95, 144]. In this thesis, we explore four works related to semantic-aware image analysis: we first use fully-connected conditional random fields to segment man-made scenes, second we smooth image using semantic labeling as structure prior, third we co-segment the shared object from a pair of images for both seen and unseen objects, finally, we localize common objects using common component activation map.

1.1 Motivation

Image analysis has been a central research area in computer vision. The ultimate objective is to be able to label each pixel of an image with regard to high-level semantics, *e.g.* the things, the stuff, and their spatial, functional and semantic relations. The image analysis includes multiple tasks, such as classification (identifying the classes of the objects), detection (localizing the objects in the image), and segmentation (assigning every pixel a semantic label).

Image classification aims to indicate which objects appear in the image from a set of known object categories, as shown in Fig. 1.1. Since spatial information is not required, image classification methods are trained using large scale training image datasets with corresponding image-level annotations that indicate the object categories present in the images. Image classification performs image-level image analysis. Recently, the deep learning based method [46] has surpassed the human-level performance on the ImageNet challenge [26].

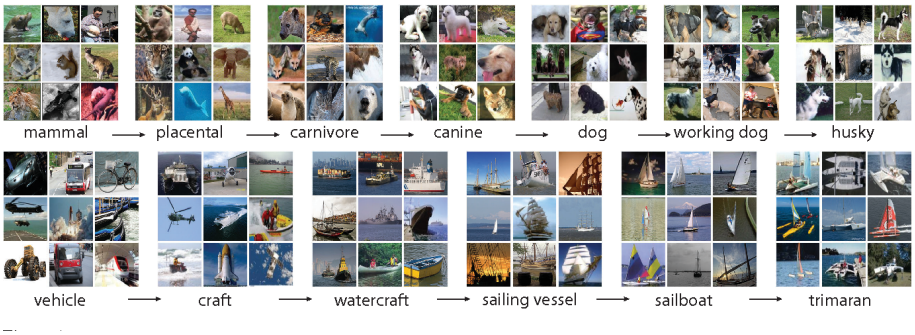

Figure 1.1: Image classification examples for images from ImageNet [26]

Object detection is the task of recognizing objects in an image and drawing bounding boxes around them (see Fig. 1.2). Detecting objects present in an image is a further step compared with classifying images for image analysis. Recently, most of object detection methods are region or proposal based, such as R-CNN [41], Fast R-CNN [40], and Faster R-CNN [101]. We can treat object detection as a region-level image analysis task.

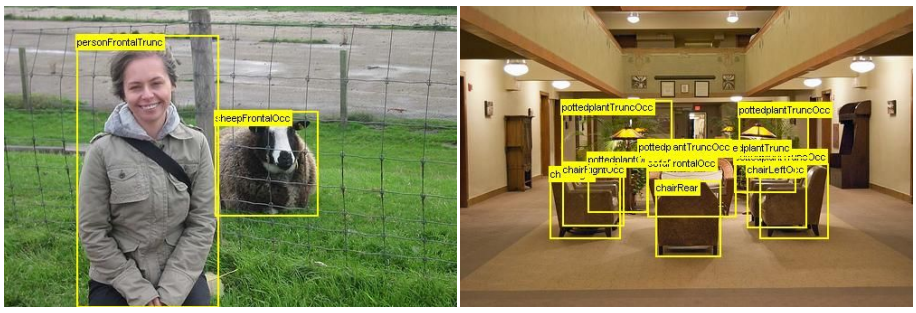

Figure 1.2: Object Detection examples for images from Pascal VOC [33].

Semantic segmentation aims to label pixels in an image from a set of predefined semantic categories, such as *person, grass, tree, sea, boat*, and *train*, as shown in Fig. 1.3. This task provides high-level semantic information about the whole image. Semantic segmentation can be formulated as two subtasks: recognition and reorganization. We can treat semantic segmentation as pixel-level image analysis.

Figure 1.3: Semantic segmentation examples for images from COCO Stuff [14].

Recently, [64] propose a new image analysis task, **panoptic segmentation**, as shown in Fig. 1.4. The goal is to unify the tasks of semantic segmentation and instance segmentation. In this task, each pixel of an image is assigned a semantic label and an instance ID. All pixels with the same semantic label and ID belong to the same *things*, *i.e.* countable objects such as animals and tools. For *stuff* classes, the instance ID is ignored, where *stuff* includes amorphous regions with similar texture or material, such as grass, sky, and road.

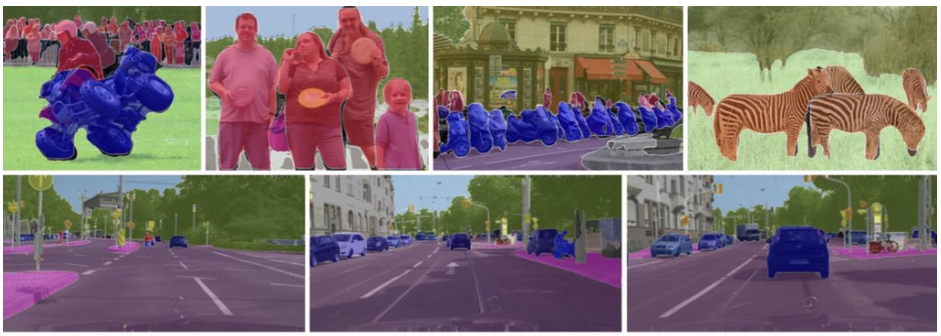

Figure 1.4: Panoptic Segmentation examples for images from [63].

1.2 Challenges

In this thesis, we focus on semantic-aware image analysis. The high-level semantic information can be used for robotics, self-driving car, healthcare, image editing, and fashion applications. However, image analysis involves many challenges when bringing semantic reasoning to real-world applications.

Irregular Structure: Semantic segmentation of man-made scenes is one of the fundamental problems in photogrammetry and computer vision. Man-made scenes, *e.g.* a street scene, as shown in Fig. 1.5, may be one of the most common scenes in daily life. These scenes exhibit strong contextual and structural information in the form of spatial interactions among components, which include buildings, doors, pavements, roads, windows or vegetation. The eTRIMS [67] is one popular image dataset for semantic segmentation of man-made scenes, which have *irregular* facades and do not follow strong architectural principles.

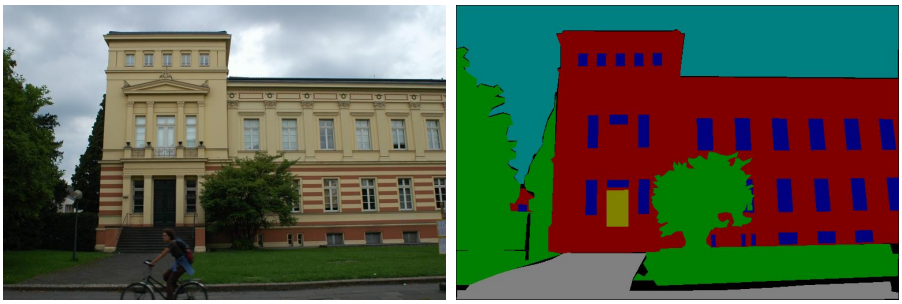

Figure 1.5: Semantic segmentation example of man-made scenes, The images are from eTRIMS dataset [67].

High-for-Low: Structure-preserving image smoothing is one of the fundamental problems in image processing and computational photography. The purpose of image smoothing is to reduce unimportant image texture or noise while preserving image structures simultaneously, as shown in Fig. 1.6. The main challenge of image smoothing is how to obtain and exploit the structural or the edge prior information to distinguish semantically pointless texture or noise from meaningful image structure.

Figure 1.6: Image smoothing examples for the input image from MSRC [117].

Segmenting Common Objects: It is a challenging task to segment common objects from the same class with large variability in terms of scale, appearance, pose, viewpoint and background, see Fig. 1.7. While image segmentation has received great attention with the recent rise of deep learning, the related task of object co-segmentation remains largely unexplored by newly developed deep learning techniques. Most of the recently proposed object co-segmentation methods still rely on models without feature learning.

Figure 1.7: Object co-segmentation examples.

Localizing Common Objects: Object co-localization is the task of simultaneously localizing objects of the same class across a group of distinct images, see Fig. 1.8. In a real-world setting, the input images are typically characterized by large scales of intra-class variation, inter-class diversity, and annotation noise. How to effectively use CNN features for this task is still not clear.

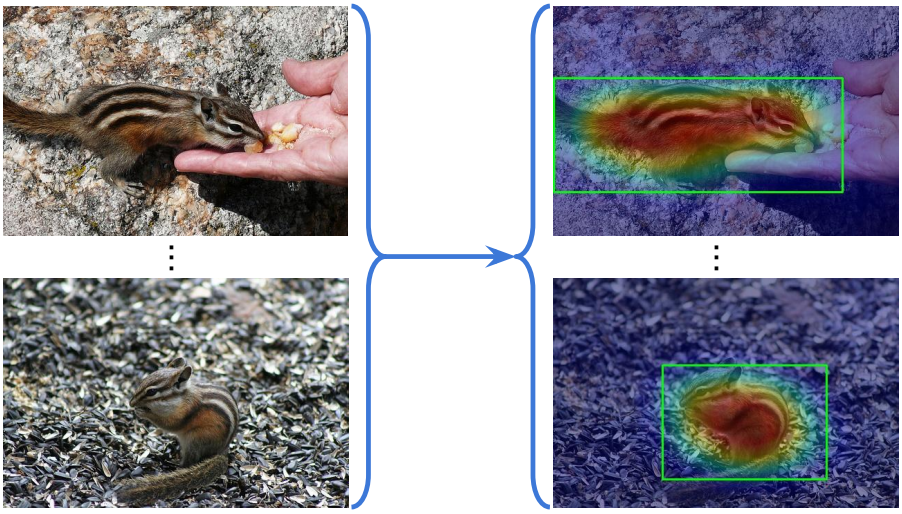

Figure 1.8: Object co-localization examples.

1.3 Contributions

The main contributions of this thesis are:

- We explore the semantic segmentation of man-made scenes using fully connected conditional random fields (CRFs). Images of man-made scenes display strong contextual dependencies in the spatial structures. Fully connected CRFs can model long-range connections within the image of man-made scenes and make use of contextual information of scene structures. The pairwise edge potentials of fully connected CRF models are defined by a linear combination of Gaussian kernels. Using the filter-based mean field algorithm, the inference is very efficient. Our experimental results demonstrate that fully connected CRF performs better than previous approaches on the eTRIMS and LabelMeFacade dataset.

- We introduce a new semantic-aware image smoothing method. Structure-preserving image smoothing aims to extract image structure from textures and noises. Recently, semantic segmentation has achieved significant progress and has been widely used in many computer vision tasks. We present an interesting observation, *i.e.* high-level semantic image labeling information can provide a meaningful structure prior naturally. Based on this observation, we propose a simple yet effective method, which we term *semantic smoothing*, by exploiting the semantic information to accomplish semantic structure-preserving image smoothing. We show that our approach outperforms the state-of-the-art approaches in texture removal by considering the semantic information for structure preservation.

- We present a deep object co-segmentation (DOCS) approach for segmenting common objects of the same class within a pair of images. This means that the method learns to ignore common, or uncommon, background *stuff* and focuses on common *objects*. If multiple object classes are presented in the image pair, they are jointly extracted as the foreground. To address this task, we propose a CNN-based Siamese encoder-decoder architecture. The encoder extracts high-level semantic features of foreground objects, a mutual correlation layer detects the common objects, and finally, the decoder generates the output foreground masks for each image. To train our model, we compile a large object co-segmentation dataset consisting of image pairs from the PASCAL dataset with common objects masks. We evaluate our approach on commonly used datasets for co-segmentation tasks and observe that our approach consistently outperforms competing methods, for both seen and unseen object classes.

- We propose an approach to localize common objects from novel object categories in a set of images. We solve this problem using a new common component activation map (CCAM) in which we treat the class-specific activation maps (CAM) as components to discover the common components in the image set. We show that our approach can generalize on novel object categories in our experiments.

1.4 Publications

The main chapters of the thesis are based on the following publications.

- **Efficient Semantic Segmentation of Man-Made Scenes using Fully Connected Conditional Random Field.** Weihao Li, Michael Ying Yang. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS), 2016. [85]

- **Semantic-Aware Image Smoothing.** Weihao Li, Omid Hosseini Jafari, Carsten Rother. International Symposium on Vision, Modeling, and Visualization (VMV), 2017. [84]

- **Deep Object Co-Segmentation.** Weihao Li*, Omid Hosseini Jafari*, Carsten Rother. Asian Conference on Computer Vision (ACCV), 2018. (*equal contribution) [82]

  Declaration: In this work, I proposed the original idea. The experimental setup, and corresponding implementations, were designed in joint discussions of me and Omid. We conducted the experiments jointly, and also wrote the paper together.

- **Localizing Common Objects Using Common Component Activation Map.** Weihao Li, Omid Hosseini Jafari, Carsten Rother. CVPR Workshop on Explainable AI, 2019. [83]

During the work on this thesis, we have also contributed to the following papers.

- **Graph Convolutional Networks Meet Markov Random Fields: Semi-Supervised Community Detection in Attribute Networks**. Di Jin, Ziyang Liu, Weihao Li, Dongxiao He, Weixiong Zhang. AAAI Conference on Artificial Intelligence, 2019. (**oral presentation**) [56]

- **Incorporating Network Embedding into Markov Random Field for Better Community Detection.** Di Jin, Xinxin You, Weihao Li, Dongxiao He, Peng Cui, Françoise Fogelman-Soulié, Tanmoy Chakraborty. AAAI Conference on Artificial Intelligence, 2019. (**oral presentation**) [57]

1.5 Outline

The remaining part of this thesis is structured as follows: In Chapter 2, we explore the topic of semantic segmentation for man-made scenes using fully connected conditional random fields. In Chapter 3, we introduce an image smoothing method by exploiting the semantic information to accomplish structure-preserving image smoothing. In Chapter 4, we present a novel deep object co-segmentation approach for segmenting common objects of the same class within a pair of images. In Chapter 5, we propose an approach to localize common objects from novel object categories in a set of images. We discuss the conclusions of this thesis in Chapter 6.

Chapter 2

Semantic Segmentation of Man-made Scenes

In this chapter, we explore the topic of semantic segmentation for man-made scenes using fully connected Conditional Random Fields (CRFs). Images of man-made scenes display strong contextual dependencies in the spatial structures. Fully connected CRFs can model long-range connections within the image of man-made scenes and make use of contextual information of scene structures. The pairwise edge potentials of fully connected CRF models are defined by a linear combination of Gaussian kernels. Using the filter-based mean field algorithm, the inference is very efficient. Our experimental results demonstrate that the fully connected CRFs perform better than previous approaches on the eTRIMS dataset.

2.1 Introduction

Semantic segmentation of man-made scenes is one of the fundamental problems in photogrammetry and computer vision. Man-made scenes, e.g. street scene, (as shown in Fig. 1.5), may be the most familiar scenes in daily life. Applications of man-made scene interpretation include 3D city modeling, vision-based outdoor navigation, and intelligent parking. Man-made scenes exhibit strong contextual and structural information in the form of spatial interactions among components, which may include buildings, cars, doors, pavements, roads, windows or vegetation. The eTRIMS [67] and LabelMeFacade [37, 13] are two

popular image dataset for man-made scene semantic segmentation, which have irregular facades and do not follow strong architectural principles. In this chapter, we will explore semantic segmentation of this kind of man-made scenes using fully connected conditional random fields.

Conditional random field (CRF) [77, 103, 117] is a popular method for modeling the spatial structure of images in semantic segmentation problem. The key idea of the semantic segmentation is to combine the low-level pixel object classifiers information and spatial contextual information within a CRF model, then running a maximum a posteriori (MAP) or maximum posterior marginal (MPM) inference method to obtain the segmentation results. However, low-connected standard (*e.g.* 4-connected or 8-connected) CRF works on a local level and cannot model the long-range dependencies of the images, so the object boundaries of these results are excessive smoothing.

CRFs with higher-order potentials, such as P^n Potts model [65] and hierarchical CRF [75, 141], have been proposed to improve semantic segmentation accuracy by enforcing label consistency in image segments (or superpixels). Both P^n Potts model and hierarchical CRF are based on unsupervised image segmentation, which is used to compute the segments or superpixels, *e.g.* normalized cuts [116], mean shift [22] and SLIC [1]. However, accurate unsupervised image segmentation is still an unsolvable problem. Segment-based P^n Potts model and hierarchical CRF model are limited by the accuracy of these unsupervised image segmentation. Mistakes in the initial unsupervised image segmentation cannot be recovered in the inference step if regions cross multiple object classes.

Recently, the fully connected CRF [69] gains popularity in the semantic segmentation problems. Fully connected CRF establishes pairwise potentials on all pairs of pixels in the image and has the ability to model long-range connections and capture fine edge details within the image. In contrast with local-range CRFs [103, 117], which are solved by an expensive discrete optimization problem [60], mean field approximation inference for the fully connected CRF is much more efficient [69]. In this chapter, we propose to use fully connected CRFs to model semantic segmentation of man-made scene problem and demonstrate it leads to state-of-the-art results.

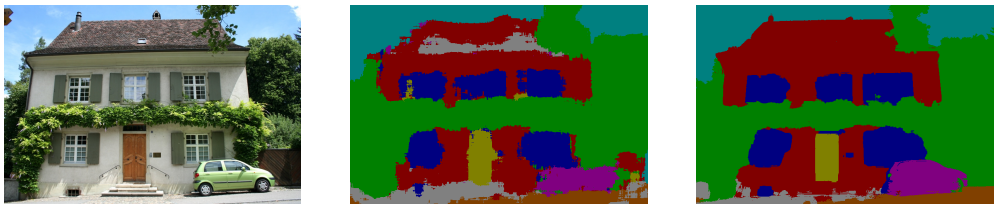

Figure 2.1: The pipeline of semantic segmentation of man-made scenes.

The whole pipeline of our system consists of two parts, as shown in Fig. 2.1: first, we train the Textonboost [117] as the unary classifier for each pixel; second, we run mean-field inference [69] for fully connected CRF to obtain maximum posterior marginal (MPM) results. Surprisingly, we find that the experimental results based fully connected CRF are better or more efficient than all previous approaches which are based on low connected CRFs on eTRIMS dataset [67].

2.2 Related Work

Man-made scene semantic segmentation approaches can be generally classified into two categories. One class methods are based on multi-class classifiers, e.g. randomized decision forest and boosting, for pixel or superpixels, then use CRFs or unsupervised segmentation methods to refine the classification results. These methods often are called as the bottom-up method, such as [141, 92, 52].

Another class of facade labeling method is shape grammar [124, 91], which is called as the top-down approach, The shape grammar methods represent the facede using a parse tree and compute a grammar by a set of production rules. However these methods are not pixel-wise labeling and not suitable for irregular man-made scene images, such as, eTRIMS [67] and LabelMeFacade [37, 13] datasets.

In [142, 141], a hierarchical CRF model is proposed to solve man-made scene images semantic segmentation problem. In this hierarchical CRF model, multi-scale mean shift algorithm [22] is used to segment the images into multi-scale superpixels. The unary potentials are the probability results of a random-

ized decision forest classifier, and then the spatial and hierarchical structures of the superpixels are connected as pairwise potentials. However, the superpixel-based hierarchical CRF model is limited by the accuracy of unsupervised image segmentation. Mistakes in the initial unsupervised segmentation cannot be recovered in the inference step if superpixels cross multiple object classes.

[92] presents a three-layered approach for semantic segmentation of building facades and man-made scenes. In the first layer, they train a recursive neural network [119] to get label probabilities of superpixels, which are got by over-segmenting the input image using mean shift algorithm [22], as the unary potentials. In the second layer, using a grid CRF model to merge initial labeling and specialized object detectors [28]. In the third layer, weak architectural principles are used as a post-processing step. However, the accuracy of the three-layered method is also restricted by the precision of unsupervised image segmentation, similar as [141].

The system of [52] uses a sequence of boosted decision trees, which are stacked using Auto-context features [126] and trained using the stacked generalization. They construct a CRF which is a pairwise 8-connected Potts model and unary classifiers are obtained directly from the image, detection, and auto-context features. Their inference method is alpha expansion [9], which costs about 24 seconds for an image on average. In contrast, the filter-based mean field approximation inference of fully-connected CRF only costs about 1 second per image of eTRIMS dataset. Therefore, fully connected CRF is much more efficient.

[37] presented a man-made scene image labeling method, which is using a random decision forest classifier and local features. Their method uses an unsupervised segmentation, *e.g.* mean shift, to refine the classification results.

Convolutional patch networks, which is a kind of convolutional neural networks (CNNs), is presented by [13]. Since both the eTRIMS [67] and LabelMe-Facade [37, 13] image databases are relatively small, this limit the classification and labeling ability of the convolutional patch networks.

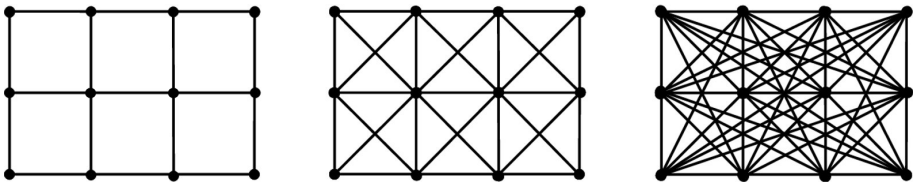

Figure 2.2: Fully-connected CRFs establish pairwise potentials on all pairs of pixels in the image and has the ability to model long-range connections. From left to right: Four-Connected CRF, Eight-Connected CRF, and Fully-Connected CRF.

2.3 Method

Conditional random field (CRF) [77, 103, 117] is a popular method for modeling the spatial structure of images in semantic segmentation problem. The CRFs model can combine the low-level pixel object classifiers information and spatial contextual information. Fully connected CRF [69] establishes pairwise potentials on all pairs of pixels in the image and has the ability to model long-range connections, as shown in Fig. 2.2, and capture fine edge details within the image. Each pairwise term of fully connected CRF is defined as a linear combination of Gaussian kernels. In contrast with local-range CRFs [103, 117], which are solved by an expensive discrete optimization problem [60], mean field approximation inference for the fully-connected CRF is much more efficient [69]. The whole pipeline of our system consists of two parts, as shown in Fig. 2.1: first, we train the Textonboost [117] as the unary classifier for each pixel independently; second, we run filter-based mean-field approximation inference [69] for fully connected CRF to obtain maximum posterior marginal (MPM) results.

2.3.1 Fully-Connected CRF

We define a random field $\mathbf{X}$ over a set of variables $\{X_1, ..., X_N\}$ which is conditioned on pixels $\{I_1, ..., I_N\}$ of a man-made scene image $\mathbf{I}$. Each random variable X_j takes a label value from the label sets $\mathcal{L} = \{l_1, ..., l_L\}$, *i.e.* X_j is the label of pixel I_j. The conditional random field is defined as a Gibbs distribution

$$P(\mathbf{X} = \mathbf{x}|\mathbf{I}) = \frac{1}{Z(\mathbf{I})} \exp(-E(\mathbf{x}|\mathbf{I})) \tag{2.1}$$

where $E(\mathbf{x})$ is the corresponding energy of the labeling $\mathbf{x} \in \mathcal{L}^N$ conditioned on $\mathbf{I}$. $Z(\mathbf{I})$ is the partition function. For convenience, we drop the conditioning on $\mathbf{I}$ in the notation. In the fully connected pairwise CRF model, the corresponding energy function is given by

$$E(\mathbf{x}) = \sum_i \phi_i(x_i) + \sum_{i<j} \phi_{ij}(x_i, x_j) \tag{2.2}$$

where i and j range from 1 to N. The unary potential $\phi_i(x_i)$ is the cost computed for pixel i taking the label x_i by a classifier given image features, and the pairwise energy potential $\phi_{ij}(x_i, x_j)$ encourage coherence in pixels x_i and x_j when they have similar features, such as, the colour values and positions.

2.3.2 Unary Potentials

The image features used in our work include 17-dimensional filter bank responses [117], RGB color, histogram of oriented gradient (HOG) [25], SIFT [90] and pixel location information. Given these image features, we compute the unary potential $\phi_i(x_i)$ for each pixel i by a multi-class classifier that produces a probability distribution over the labeling x_i independently. The form of unary potential $\phi_i(x_i)$ is the negative log likelihood, *i.e.* corresponding probability distribution P of the labeling assigned to pixel i,

$$\phi_i(x_i) = -\log P(x_i|\mathbf{I}). \tag{2.3}$$

The unary potentials incorporate shape, texture, location, and color descriptors, which are derived from TextonBoost [117, 75]. We use the extended TextonBoost framework, which boosts classifiers defined on above-mentioned features together. The implementation used here is the Automatic Labelling Environment (ALE) [75]. The result of unary classifiers is usually noisy, as shown in the middle image of Fig. 2.1.

2.3.3 Pairwise Potentials

The pairwise potentials in fully connected CRFs model have the form

$$\phi_{ij}(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) \tag{2.4}$$

where $k^{(m)}$ is a Gaussian kernel, $w^{(m)}$ is weight of the kernel, μ is label compatibility function, and $\mathbf{f}_i, \mathbf{f}_j$ are feature vectors for pixel i and j, which are color values and pixel positions as [69].

In this chapter, we use Potts model, and $\mu(x_i, x_j) = [x_i \neq x_j]$. For man-made scene semantic segmentation we use contrast-sensitive two-kernel potentials,

$$\begin{aligned} k^{(1)}(\mathbf{f}_i, \mathbf{f}_j) &= w^{(1)} \exp(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}), \\ k^{(2)}(\mathbf{f}_i, \mathbf{f}_j) &= w^{(2)} \exp(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}). \end{aligned} \tag{2.5}$$

where I_i and I_j are the color vectors and p_i and p_j are positions. The first part $k^{(1)}(\mathbf{f}_i, \mathbf{f}_j)$ is the smoothness kernel, which helps to remove small isolated regions, and the second part $k^{(2)}(\mathbf{f}_i, \mathbf{f}_j)$ is the appearance kernel, which encourages nearby pixels to have the same labels when they have similar color. The degrees of similarity are controlled by parameters θ_α, θ_β and θ_γ.

2.3.4 Inference

Following [69], we use a mean field method for approximate Maximum Posterior Marginal (MPM) inference. The mean field approximation computes an alternative distribution $Q(\mathbf{X})$ over the random variables $\mathbf{X}$, instead of computing the posterior distribution $P(\mathbf{X})$ directly. Distributions $Q(\mathbf{X})$ is a product of independent marginals, *i.e.* $Q(\mathbf{X}) = \prod_i Q_i(\mathbf{X}_i)$. The mean field approximation minimizes the KL-divergence $D(Q||P)$ between distribution Q and the exact distribution P. The mean field inference performs the following message

passing iterative update until convergence:

$$\begin{aligned} Q_i(x_i = l) &= \frac{1}{Z_i} \exp\{-\phi_u(x_i) \\ &\quad - \sum_{l' \in \mathcal{L}} \sum_{j \neq i} Q_j(x_j = l') \phi_{ij}(x_i, x_j)\} \\ &= \frac{1}{Z_i} \exp\{-\phi_u(x_i) \\ &\quad - \sum_{l' \in \mathcal{L}} \sum_{m=1}^{K} \mu(l, l') w^{(m)} \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l')\} \end{aligned} \tag{2.6}$$

where Z_i is the marginal partition function of pixel i used to normalize the marginal. Updating the message passing iteration in sequence across pixels, KL-divergence will be convergence [66]. Directly computing this message passing iterative is intractable, because for each pixel, evaluating the sum of all of the other pixels is required. This is the computational bottleneck of the message passing iterative. To make this update tractable and efficient, a high dimensional Gaussian filter can be used [2, 69]. The transformation is:

$$\begin{aligned} \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l) &= \sum_{j} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l) - Q_j(l) \\ &= [G_m \otimes Q(l)](\mathbf{f}_i) - Q_j(l), \end{aligned} \tag{2.7}$$

where G_m is the corresponding Gaussian kernel of $k^{(m)}$ and $\otimes$ is the convolution filtering. Using the permutohedral lattice [2], which is a highly efficient convolution data structure, the approximate message passing can be updated in time $\mathcal{O}(Nd)$ [69]. Of cause, other filtering methods also can be used for the approximate message passing, *e.g.* domain transform filtering [39, 131].

The first smoothness kernel is a Gaussian blur. And the second appearance kernel actually is the joint or cross bilateral filtering [125, 99, 32], in which $Q(l)$ is the input image and I is the reference (or guidance) image. After running the update step in a fixed number, in this chapter, we update 10 times iteration, then we get the MPM result from the final iteration,

$$x_i \in \arg\max_l Q_i(x_i = l). \tag{2.8}$$

2.3.5 Learning

For training unary potentials, we set the parameters of the low-level feature descriptors, such as HOG, Texton, and SIFT, using the setting of Automatic Labelling Environment (ALE). For the parameters of the CRFs, we use 5 folders cross-validation to learning the weights of the unary responses and Gaussian kernels.

2.4 Experiments

We evaluate the fully-connected CRFs on all two irregular man-made scene images benchmark datasets: eTRIMS dataset and LabelMeFacade Dataset. For eTRIMS dataset, we perform a 5-fold cross-validation as in [141] mentioned by dividing 40 images into a training set and 20 images into a test set randomly. For LabelMeFacade dataset, we use the pre-separated training and testing as the same as [37, 13] mentioned. We compare our results with against [52] and [13].

2.4.1 Datasets

eTRIMS dataset [67] includes 60 man-made scene images, which are labeled with 8 classes: *building, car, door, pavement, road, sky, vegetation and window.* And each image have an accurate pixel-wise annotation. For evaluation, we perform a 5-fold cross-validation as in [141] by dividing 40 images into a training set and 20 images into a test set randomly. Then we run the experiment five times and report the average accuracy.

LabelMeFacade Dataset is presented by [37], which are also labeled with 8 classes: *building, car, door, pavement, road, sky, vegetation and window.* The images of LabelMeFacade dataset are taken from LabelMe dataset [109]. There are 945 images in the dataset, which are split as two sets, 100 images for training and 845 images for testing. Similar with the eTRIMS dataset, facades in LabelMeFacade dataset are highly irregular.

2.4.2 Results

We compare our approach with the state-of-the-art man-made scenes image segmentation methods on eTRIMS dataset and LabelMeFacade dataset, *i.e.* facade segmentation using Auto-Context [52] and Convolutional Patch Networks [13]. We choose the average, overall and intersection over union (IoU) score as the evaluation measures. Overall is the pixel-wise labeling accuracy, which is computed over the whole image pixels for all classes. Average is the pixel-wise labeling accuracy computed for all classes and the averaged over these classes. IoU is defined as $TP/(TP + FP + FN)$. TP represents the true positive, FP means false positive and FN indicates the false negative.

Class	Textonboost	CRF	AC [52]	Ours
Building	74.38	80.38	**92.50**	84.14
Car	81.58	85.88	76.60	**87.06**
Door	77.40	**80.24**	65.30	79.80
Pavement	59.02	**62.22**	48.80	60.46
Road	80.02	81.36	**82.10**	81.16
Sky	97.16	98.42	98.90	**99.32**
Vegetation	89.70	90.22	**92.90**	91.74
Window	**77.66**	75.94	68.20	71.10
Average	79.62	81.83	78.14	**81.85**
Overall	80.12	83.31	**87.29**	84.72
IoU	59.44	63.52	63.54	**64.81**

Table 2.1: The quantitative results on the eTRIMS dataset. Textonboost is trained using Automatic Labelling Environment. The CRF is a 4-connected CRF. AC is the Auto-context method. Our method is the fully connected CRF.

We show the quantitative experimental results of the eTRIMS dataset in Table 2.1. Our method outperforms the previous state-of-the-art approaches [52] on the eTRIMS dataset. We get the Average 81.85% and IoU 64.81%. Our Average and IoU are highest, and we get five classes out of eight higher than the Auto-Context method [52], which is the benchmark on the eTRIMS dataset before. Note that the Auto-Context method [52] uses detection as a preprocessing step. We do not use any detection information, and our approach is efficient, which only need about one second in the inference step. Fig. 2.3

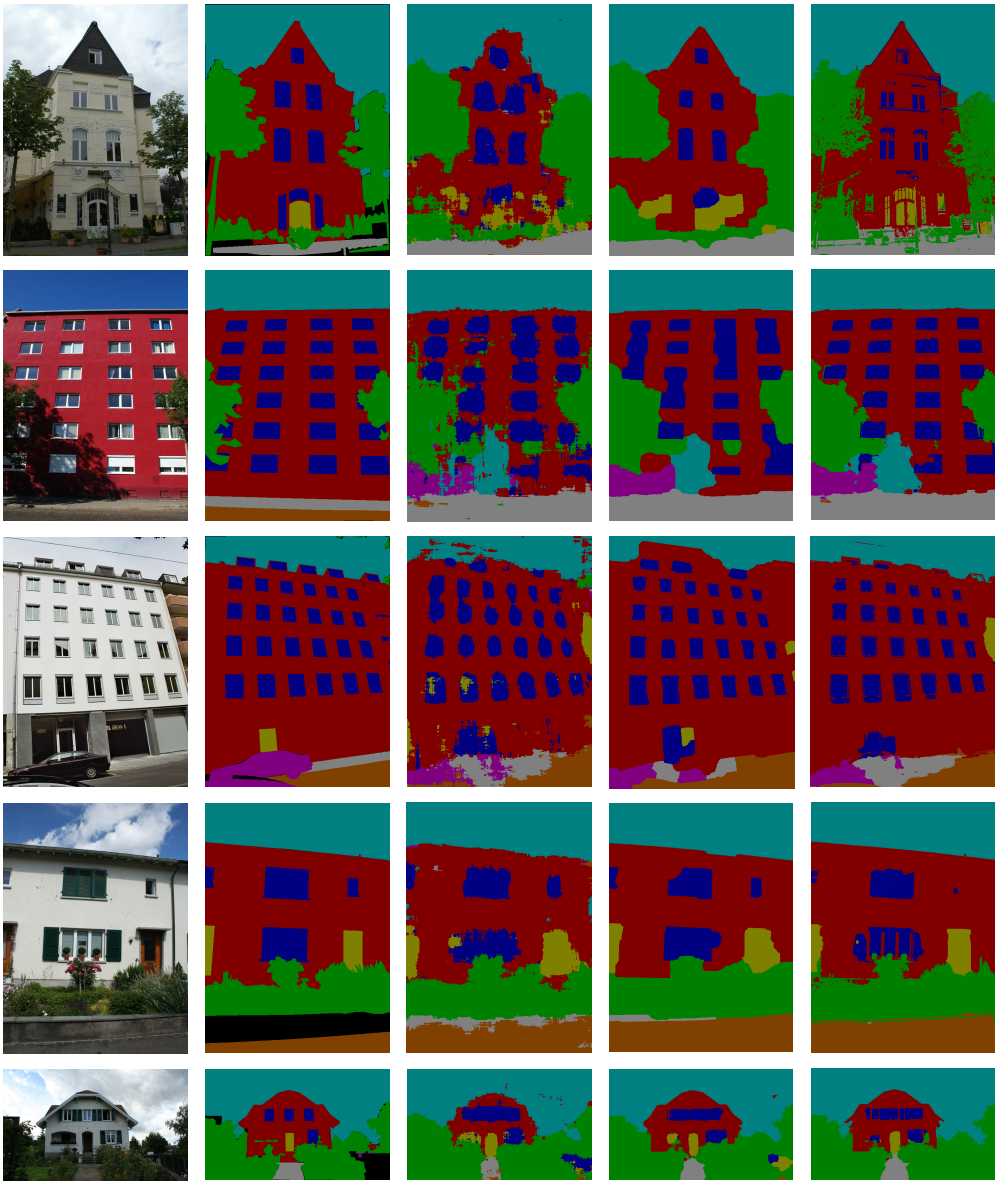

Figure 2.3: The qualitative results of the eTRIMS dataset. First column are examples of the testing images and 2nd-column are the corresponding ground truth. (3rd-column to 5th-column) man-made scene semantic segmentation results using the Textonboost classifier, the CRF model and the fully connected CRF model, respectively. The fully connected CRF model obtain more accurate and detailed results than the low connected CRF.

shows qualitative segmentation results of our method. Our method obtains more accurate and detailed results than the low connected CRF.

Class	Textonboost	CRF	AC [52]	CPN [13]	Ours
Average	**61.39**	60.95	49.04	58.98	59.53
Overall	75.47	77.40	75.23	74.33	**79.27**
IoU	46.85	48.07	39.57	-	**48.48**

Table 2.2: The quantitative results on the LabelMeFacade dataset. Textonboost is trained using Automatic Labelling Environment. The CRF is a 4-connected CRF. AC is the Auto-context method. CPN is the Convolutional Patch Networks method. Our method is the fully connected CRF.

We show the quantitative experimental results of LabelMeFacade dataset in Table 2.2.Our method outperforms the previous state-of-the-art approaches [52] and [13] on LabelMeFacade dataset. Since [13] only provide Average and Overall result, we just compare [13] with these two measures. In contrast with [52], they regard the 'various' as a class, we do not consider 'various' as a class. So the LabelMeFacade dataset has eight classes as the eTRIMS dataset. We get the Overall accuracy 79.27% and IoU accuracy 48.48%. Our Average and IoU are highest, and we get five classes out of eight higher than the Convolutional Patch Networks method [13], which is the benchmark on the eTRIMS dataset before. Fig. 2.4 shows qualitative segmentation results of our method. Our method obtains more accurate and detailed results than the low connected CRF.

2.5 Conclusion

In this chapter, we explore man-made scene semantic segmentation using fully connected conditional random fields model, which is very efficient and only need about one second in the inference step. The method outperforms the previous state-of-the-art approaches on the eTRIMS dataset and the LabelMeFacade dataset, which obtains more accurate and detailed results.

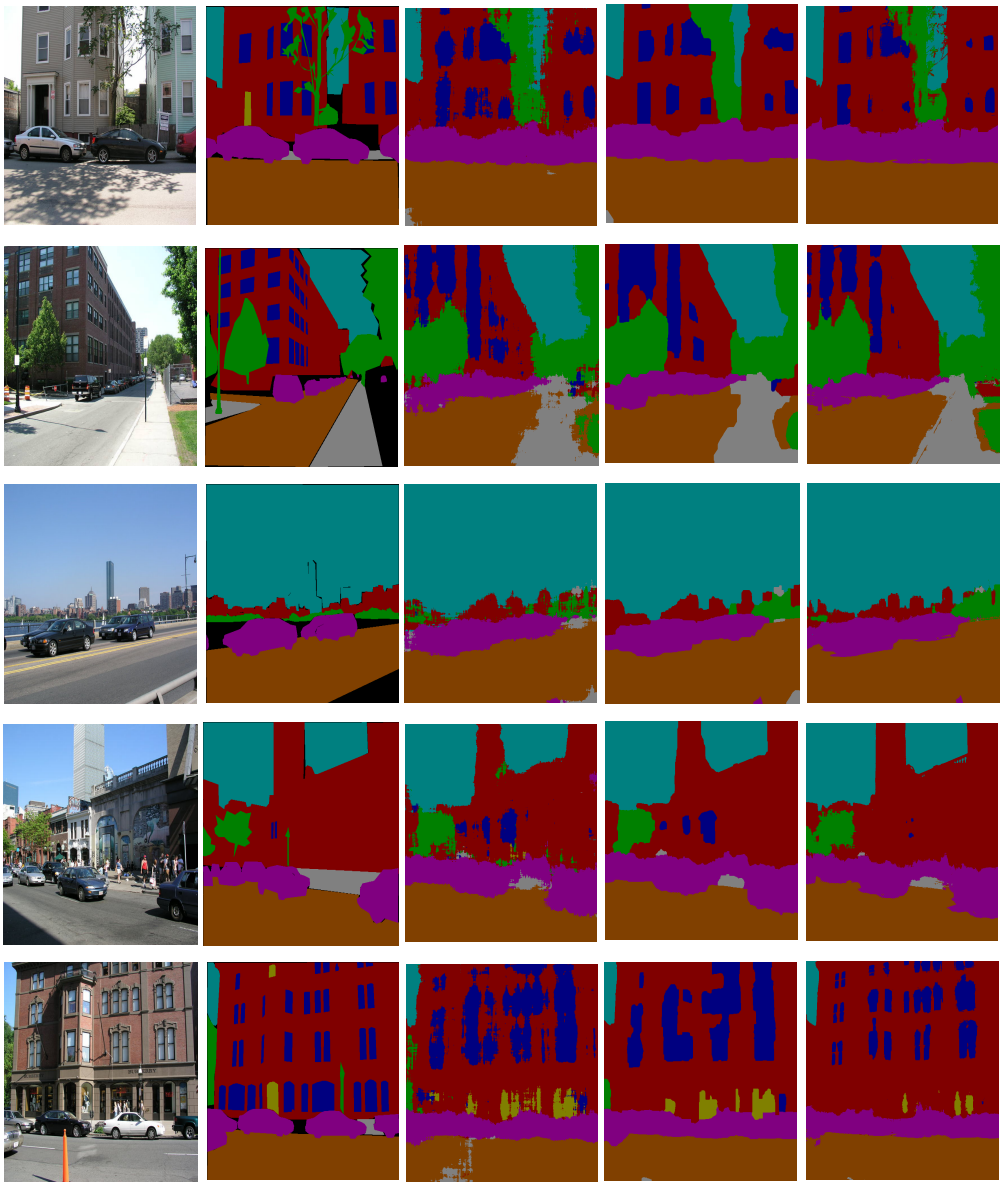

Figure 2.4: The qualitative results of the LabelMeFacade dataset. The first column is examples of the testing images and 2nd-column are the corresponding ground truth. (3rd-column to 5th-column) man-made scene semantic segmentation results using the Textonboost classifier, the CRF model and the fully connected CRF model, respectively. The fully connected CRF model obtain more accurate and detailed results than the low connected CRF.

Chapter 3

Semantic-Aware Image Smoothing

Structure-preserving image smoothing aims to extract semantically meaningful image structure from texture, which is one of the fundamental problems in computer vision and graphics. However, it is still not clear how to define this concept. On the other hand, semantic image labeling has achieved significant progress recently and has been widely used in many computer vision tasks. In this paper, we present an interesting observation, *i.e.* high-level semantic image labeling information can provide a meaningful structure prior naturally. Based on this observation, we propose a simple and yet effective method, which we term *semantic smoothing*, by exploiting the semantic information to accomplish semantically structure-preserving image smoothing. We show that our approach outperforms the state-of-the-art approaches in texture removal by considering the semantic information for structure preservation. Also, we apply our approach to three applications: detail enhancement, edge detection, and image segmentation, and we demonstrate the effectiveness of our semantic smoothing method on these problems.

3.1 Introduction

Structure/edge-preserving image smoothing [138, 61, 137] is one of the fundamental problems in image processing, computational photography, and computer vision. The purpose of image smoothing is to reduce unimportant image texture or noise while preserving semantically meaningful image structures

Figure 3.1: Semantic Smoothing on MSRC-21 dataset. In this example, image(a) contains a textured bench in a grassland. As a result, it is difficult for the state-of-the-art structure-preserving and edge-preserving smoothing methods to obtain smoothing results with accurate structure (b)-(g). (b) Domain Transform (DT) [39], (c) L_0 Smoothing [136] ($\lambda = 0.04$), (d) Rolling Guidance Filter (RGF) [147], (e) Region Covariances (RG) [61] ($k = 5$, $\sigma = 0.2$, Model 1), (f) Relative Total Variation (RTV) [138] ($\lambda = 0.005$, $\sigma = 3$) and (g) Semantic Filtering (SF) [143]. Our method effectively preserves semantically meaningful structure and smooth out detail and texture. Best viewed in color.

simultaneously [138, 143]. It has achieved widespread use in various applications, including texture removal, edge extraction, image abstraction, seam carving, and tone mapping.

The main challenge of image smoothing is how to obtain and exploit the structural or the edge prior information to distinguish semantically pointless texture or noise from meaningful image structure [138, 147, 143]. The majority of edge-preserving image filters apply low-level feature, *i.e.* image gradients, as edge prior information, such as bilateral filtering [125] and guided filter [45]. For structure-preserving image smoothing, relative total variation [138], diffusion maps [35], and region covariances [61] measures are used to separate texture from the image structure. Recently, Yang [143] use an edge detector for iterative edge-preserving texture filtering to exploit mid-level vision feature, *i.e.* structured edges. Although these methods work well for some tasks, it is not clear how to define the meaningful image structure. For example in Fig. 3.1 (b-g), it is difficult for previous approaches to preserve the bench structure when they only consider low-level and mid-level vision features of an image.

In this chapter, we present an observation, *i.e.* high-level semantic information can provide a meaningful structure prior to image smoothing naturally. Recently, semantic labeling has been heavily studied in computer vision community [117, 151, 89, 50]. Semantic information provides an object-level semantically meaningful structure prior, such as object boundaries, which help to reduce the negative effect of sharp edges inside of objects. Based on this observation, in this chapter, we present a simple and yet effective method which exploits semantic labeling information to accomplish texture removal and meaningful structure preservation. We call this new concept *semantic smoothing*. Besides utilizing high-level semantic information, our method also can combine low-level vision features, *i.e.* image appearance, and mid-level vision information, *i.e.* image edges.

Our method has two unique properties: meaningful structure preservation and interior detail removal. As shown in Fig. 3.1, input image Fig. 3.1 (a) contains a textured bench in the foreground and a grassland in the background. Current state-of-the-art image smoothing methods cannot successfully separate bench from its texture and preserve its structure as shown in Fig. 3.1 (b)-(g). Our

proposed semantic smoothing technique outperforms other approaches by preserving the bench structure effectively as illustrated in Fig. 3.1 (h). To the best of our knowledge, it is the first structure-preserving image smoothing method which exploits high-level semantic segmentation information.

The following sections are organized as follows. The related works are discussed in Section 3.2. In Section 3.3 our semantic-aware image smoothing method is described. In Section 3.4 experimental results and applications are presented.

3.2 Related Work

We categorize the related work into two aspects: image smoothing and semantic segmentation. First, we discuss edge-preserving and structure-preserving image smoothing methods. Second, we briefly review development progress of semantic segmentation and semantic information in other vision problems.

3.2.1 Image Smoothing

The image smoothing methods can be separated into two classes: edge-preserving and structure-preserving smoothing. The bilateral filter [125] is one of the most popular edge-preserving filtering methods which replaces the intensity value of each pixel in the image with a weighted average of intensity values of its neighboring pixels. In joint bilateral filters [99, 32], the range filter is applied to a guidance image from another domain. As edge-preserving image smoothing or filtering methods, we can also mention anisotropic diffusion [98], weighted least square [36], local Laplacian pyramid [97], domain transform [39], and semantic filtering [143]. However, it is hard to separate high-contrast textured regions or patterns from the meaningful structures of an image by using these edge-preserving techniques. The structure-preserving image smoothing techniques aim to separate the image structure and texture. One of the most popular structure-preserving image smoothing methods is Xu *et al.* [138], which uses the relative total variation (RTV) measure to decompose structures from textures. They first model a regularization term based on the RTV measure,

then solve a global optimization to extract the main structures and to obtain the smoothed image. Zhang *et al.* [146] first segment the input image into superpixels then they build a minimum spanning tree for each superpixel to accelerate image filtering. Shen *et al.* [115] proposes a mutual-structure joint filtering towards preserving common structures of an input and a guidance image. As other structure-preserving image smoothing techniques we can mention total variation [107], local extrema [120], structure adaptive[74], rolling guidance filter [147], and geodesic [23].

Recently, several learning-based methods have also been proposed for image filtering [137, 6]. In contrast, we exploit the semantic segmentation information as a meaningful structure prior to the semantic structure-preserving image smoothing.

3.2.2 Semantic Segmentation

Semantic segmentation is one of the key problems in image understanding. The goal of semantic segmentation is to label each pixel of the image with the class of its enclosing object. A common pipeline of semantic segmentation is first to train pixel-based classifiers, such as Textonboost [117] or fully convolutional networks (FCN) [89], then using a probabilistic graphical model, such as CRF [117, 17, 152, 78], to improve the performance by modeling structured dependencies. With the development of semantic segmentation techniques, other computer vision problems exploit high-level semantic information, such as optical flow [113, 5], depth prediction [50, 135], depth upsampling [49, 110], object attributes [130, 151], intrinsic image estimation [130], 3D reconsecration [42, 73, 76].

However, smoothing image using semantic segmentation information has not been exploited before. In this chapter, we propose a novel semantic-aware approach which exploits the semantic information for structure preserving image smoothing.

3.3 Semantic Smoothing

In this section, we introduce our semantic image smoothing method, which exploits high-level semantic information to achieve semantically meaningful structure preserving smoothing.

Given an input image $\mathbf{t}$ and its semantic labeling $\mathbf{s}$, our goal is to compute a new smoothed image $\mathbf{x}$, which is as similar as possible to the input image $\mathbf{t}$ while preserving the semantically meaningful image structure and reducing the texture or noise. We model our *semantic smoothing* as an energy minimization problem. Formally, we define the energy function as a weighted sum of two energy terms

$$E(\mathbf{x}) = E_d(\mathbf{x}; \mathbf{t}) + E_r(\mathbf{x}; \mathbf{t}, \mathbf{s}), \tag{3.1}$$

where E_d is the data term and E_r is the regularization term.

3.3.1 Data Term

The purpose of the data term is to minimize the distance between the input image $\mathbf{t}$ and the smoothed image $\mathbf{x}$. Without this data term, there will be a trivial solution where all of the pixels will be assigned to the same color value. We define the data term E_d as

$$E_d(\mathbf{x}; \mathbf{t}) = \sum_i (x_i - t_i)^2, \tag{3.2}$$

where i is the pixel index. With this term, smoothed image $\mathbf{x}$ will be limited within a range around the input image $\mathbf{t}$.

3.3.2 Regularization Term

The regularization term E_r strive to achieve smoothness by jointly considering the low-level appearance, the mid-level edge, and the high-level semantic information. The regularization term E_r is defined as

$$E_r(\mathbf{x}; \mathbf{t}, \mathbf{s}) = \sum_i \sum_{j \in \mathcal{N}(i)} W_{i,j}(x_i - x_j)^2, \tag{3.3}$$

where $\mathcal{N}(i)$ is a set of neighboring (four or eight) pixels around the pixel i and the weight $W_{i,j}$ represents the similarity between the pixel i and the pixel j.

Our $W_{i,j}$ consists of three potential functions and is defined as

$$W_{i,j} = \lambda_a w^a_{i,j} + \lambda_e w^e_{i,j} + \lambda_s w^s_{i,j}, \tag{3.4}$$

where the first factor $w^a_{i,j}$ is the appearance potential which is used to control the low-level information. The second factor $w^e_{i,j}$ is based on the edge detection and is used to control the mid-level information. The last factor $w^s_{i,j}$ is the semantic potential which exploits the high-level semantic information. The weights λ_a, λ_e, and λ_s are used to control the effect of the low-level, the mid-level and the high-level information on the final smoothed output, respectively. These three parts are explained in detail below.

3.3.2.1 Appearance potential

The appearance potential $w^a_{i,j}$ of the pixel i and the pixel j is defined as

$$w^a_{i,j} = \exp(-\frac{||\mathbf{f}_i - \mathbf{f}_j||^2}{\sigma_a}), \tag{3.5}$$

where $\mathbf{f}_i$ and $\mathbf{f}_j$ are three-dimensional vectors representing the Lab color values of the pixel i and the pixel j and σ_a is a range parameter.

We use the appearance potential to measure the difference of the low-level vision feature, *i.e.* color, between the pixel i and the pixel j. In this setting, neighboring pixels of the input image with similar colors are assigned to larger weights and neighboring pixels with different colors are assigned to smaller weights.

3.3.2.2 Edge potential

The edge potential $w^e_{i,j}$ between the pixel i and the pixel j is defined as

$$w^e_{i,j} = \exp(-\frac{\beta^2_{i,j}}{\sigma_e}), \tag{3.6}$$

where $\beta_{i,j} \in [0, 1]$ is the boundary strength measure between the pixel i and the pixel j and σ_e is a range parameter.

Recently, Yang [143] uses an edge detector [29] for edge-preserving image filtering. In contrast, we utilize image edges as the mid-level vision cue to help the appearance potential and the semantic potential. In this work, we use the structured edge detector [29] to calculate boundary strength measure $\beta_{i,j}$.

3.3.2.3 Semantic potential

The semantic potential is the key part of our semantic smoothing. Based on the semantic labeling $\mathbf{s}$, the semantic potential between the pixel i and the pixel j can be written as

$$w_{i,j}^{s} = \begin{cases} \gamma_{high} & \text{if } s_i = s_j \\ \gamma_{low} & \text{otherwise,} \end{cases} \tag{3.7}$$

where s_i and s_j present semantic labeling of the pixel i and the pixel j. γ_{high} and γ_{low} are weight parameters and $\gamma_{high} > \gamma_{low}$. When neighboring pixels i and j have the same semantic labeling, we assign a larger weight to encourage these two pixels to have close color values in the output smoothed image. In contrast, when neighboring pixels i and j have different semantic labeling, they are assigned a smaller weight. For each class label, it is possible to set different γ_{high} values to control the different smoothing strength. In this work, for simplicity, we set γ_{high} to 1.0 for all semantic classes and we set γ_{low} to zero. Semantic information help to reduce the adverse effect of the object's interior sharp edges and texture.

3.3.3 Optimization

The objective function in Equation 3.1 is strictly convex and can be written in a matrix and vector form as

$$E(\mathbf{x}) = (\mathbf{x} - \mathbf{t})^{\mathsf{T}}(\mathbf{x} - \mathbf{t}) + \mathbf{x}^{\mathsf{T}}\mathbf{A}\mathbf{x} \tag{3.8}$$

where matrix $\mathbf{A}$ is a Laplacian matrix which is defined as

$$\mathbf{A} = \mathbf{D} - \mathbf{W}, \tag{3.9}$$

where $\mathbf{W}$ is an adjacency matrix $\{W_{i,j} \| j \in \mathcal{N}(i)\}$ and $\mathbf{D}$ is a degree matrix which is defined as

$$D_{i,j} = \begin{cases} \sum_{j \in \mathcal{N}(i)} W_{i,j} & i = j \\ 0 & i \neq j. \end{cases} \tag{3.10}$$

By setting the gradient of $E(\mathbf{x})$ defined as in Equation 3.8 to zero, the final smoothing result $\mathbf{x}$ is obtained by solving the linear system based on a large sparse matrix:

$$(\mathbf{I} + \mathbf{A})\mathbf{x} = \mathbf{t} \tag{3.11}$$

where $\mathbf{I}$ is an identity matrix.

3.4 Experimental Results and Applications

Our semantic smoothing method can benefit several image editing and manipulation applications due to its special properties, *i.e.* meaningful structure preservation and interior detail removal. In this section, first, we introduce the datasets which we used in our experiments. Second, we visually compare the texture removal results of our proposed semantic smoothing approach with the state-of-the-art methods. Finally, to show the effect of our approach, we apply it to three applications: detail enhancement, edge detection, and image segmentation.

3.4.1 Datasets

MSRC-21 dataset [117] consists of 591 color images with the following 21 object classes, such as grass, tree, cow, sheep, water and so forth. Cimpoi [20] also use MSRC-21 dataset for texture recognition and segmentation task. In order to ensure proportional contributions from each class approximately, the dataset is split into 45% training, 10% validation and 45% test images. We use the standard split of the dataset from [117] to train the textonboost [117], which incorporates shape, texture, location, and color descriptors. Then, we use the trained textonboost to obtain the semantic segmentation. Lastly, we apply the dense CRF [69] to refine the semantic segmentation results and we use this refined version as high-level semantic information input to our smoothing approach. The run-time for each image is roughly 2.0 seconds.

PASCAL VOC dataset [33] consists of one background class and 20 foreground object classes including person, bird, cat, cow, dog and so forth. There are 1464 images for training, 1449 for validating and 1456 for testing, respectively. Recently, the fully convolutional network (FCN) [89] is mainly utilized for estimating the semantic segmentation on PASCAL VOC dataset. Also in this work, we employ the publicly available pre-trained FCN [89] for obtaining the semantic labeling for PASCAL VOC. Then, we use the dense CRF [69] to refine the FCN results for using it as the input to our semantic smoothing. The run-time for each image is roughly 2.8 seconds.

3.4.2 Texture Removal

Texture removal, which is also called as texture smoothing, aim to separate the meaningful structures from textures. We compare our semantic smoothing results with the state-of-the-art image smoothing techniques, such as Relative Total Variation (RTV) [138] and Semantic Filtering (SF) [143]. We use the authors' publicly available implementations. It is difficult to quantitatively evaluate image smoothing methods, therefore similar to most of the state-of-the-art methods [143, 138], we present the visual comparison evaluation in Fig. 3.2, Fig. 3.3 and Fig. 3.4. We visually compare our proposed semantic smoothing technique with [138, 143] on MSRC-21 dataset (see Fig. 3.2 and Fig. 3.3) and PASCAL VOC dataset (see Fig. 3.4). As illustrated in these figures, our semantic-aware image smoothing performs better in terms of preserving meaningful structures and reducing object interior textures. For instance, if we look at the black cow in the first row of Fig. 3.2, there are strong edges inside of the cow's body in other approaches' results, while our approach is able to remove these semantically meaningless edges.

3.4.3 Applications

3.4.3.1 Detail Enhancement

Detail enhancement aims to increase the visual appearance of images, which is widely used in image editing. Thanks to the property of structure-preserving

(a) Images

(b) SF

(c) RTV

(d) Ours

Figure 3.2: Visual comparison of texture removal results on MSRC dataset. (a) input images, (b) Semantic Filtering (SF) [143], (c) Relative Total Variation (RTV) [138] and (d) Our semantic smoothing results. Best viewed in color.

(a) Images

(b) SF

(c) RTV

(d) Ours

Figure 3.3: Visual comparison of texture removal results on MSRC dataset. (a) input images, (b) Semantic Filtering (SF) [143], (c) Relative Total Variation (RTV) [138] and (d) Our semantic smoothing results. Best viewed in color.

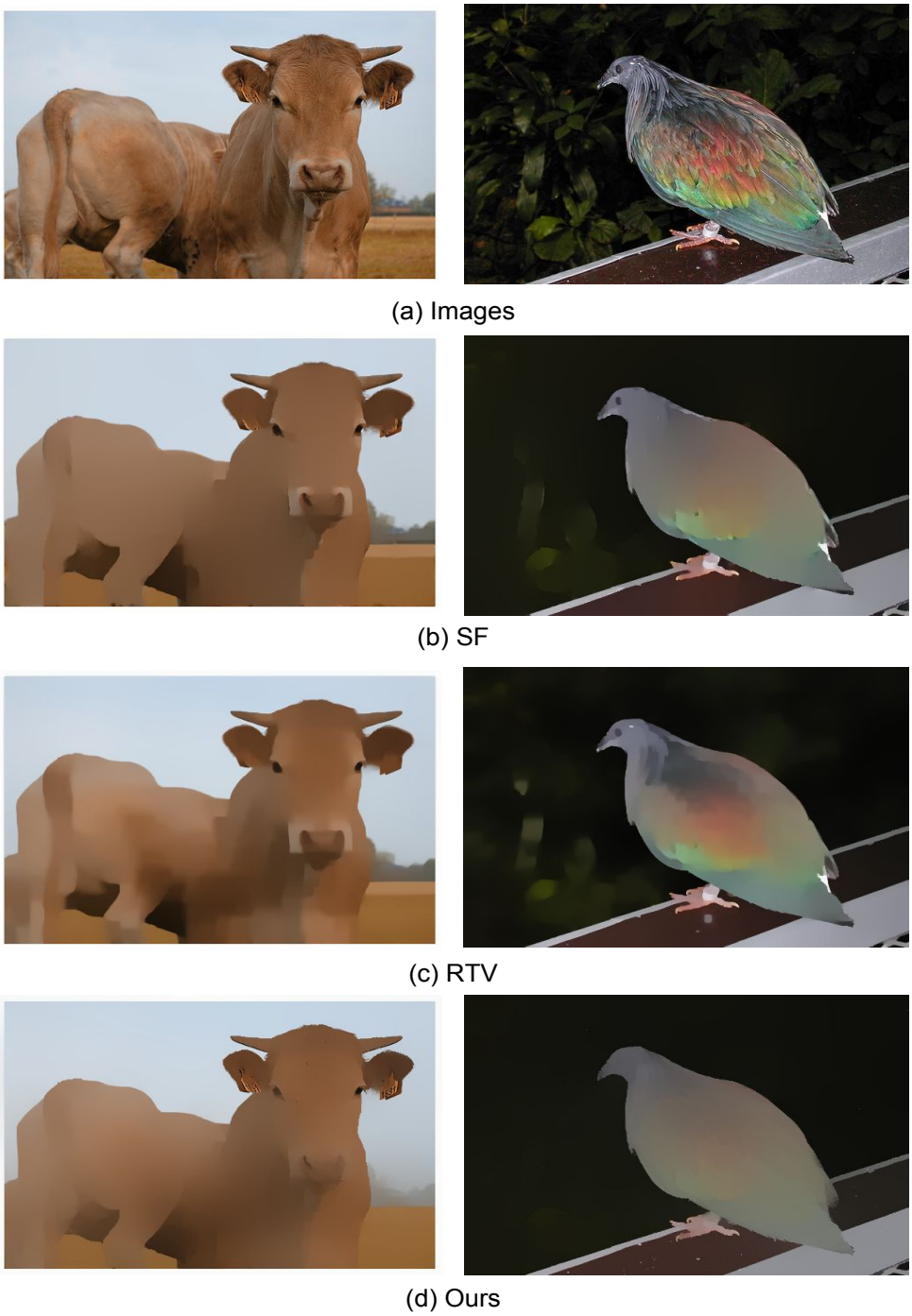

Figure 3.4: Visual comparison of texture removal results on PASCAL VOC dataset (a) input images, (b) Semantic Filtering (SF) [143], (c) Relative Total Variation (RTV) [138] and (d) Our semantic smoothing results. Best viewed in color.

image smoothing, *i.e.* structure-texture decomposition, we can apply our semantic smoothing method to enhance the underlying details or textures of an image. First, we use our semantic smoothing method to decompose the input image into structures and details. Then we add the details back to the input image. That means we augment the contrast in detail components of the input image.

Fig. 3.5 shows two examples. Given two input images Fig. 3.5 (a) and (e) and their semantic smoothing results Fig. 3.5 (b) and (f), we can decompose the texture information Fig. 3.5 (c) and (g) and obtain the detail enhancement results Fig. 3.5 (d) and (h). Since our smoothing method can effectively preserve the object-level structure and remove object interior edges, it can effectively enhance the underlying detail, particularly *interior* texture and edges of objects, without blurring the main structure of objects.

3.4.3.2 Edge Detection

Edge detection is one of the challenging tasks in computer vision for a long time. The purpose of edge detection is to extract visually salient edges or object boundaries from the input image. Boundary and edge can be used in a broad range of computer vision or graphics tasks, such as semantic segmentation, object recognition, image editing, and tone mapping. Our method can be applied to object-level edge extraction thanks to its ability to preserve semantically meaningful structures and remove many unimportant details, such as interior edges of the object especially.

Fig. 3.6 (a) shows an input image in grass texture with a salient foreground, *i.e.* a cow. Since the texture has high contrast, applying the Canny edge detector [15] cannot produce reasonable results directly from the input image, see Fig. 3.6 (c). Structured edge detection [29] is a popular edge detection method based on random forests, which can detect salient edges. It achieves better results as demonstrated in Fig. 3.6 (e) and thinned edges Fig. 3.6 (g), which is obtained by standard non-maximal suppression technique. We can see that some of the detected edges come from the textures. In contrast, our method first produces an object-level structure-preserving smoothed image, which removes insignificant details as Fig. 3.6 (b). We can improve the result of these edge

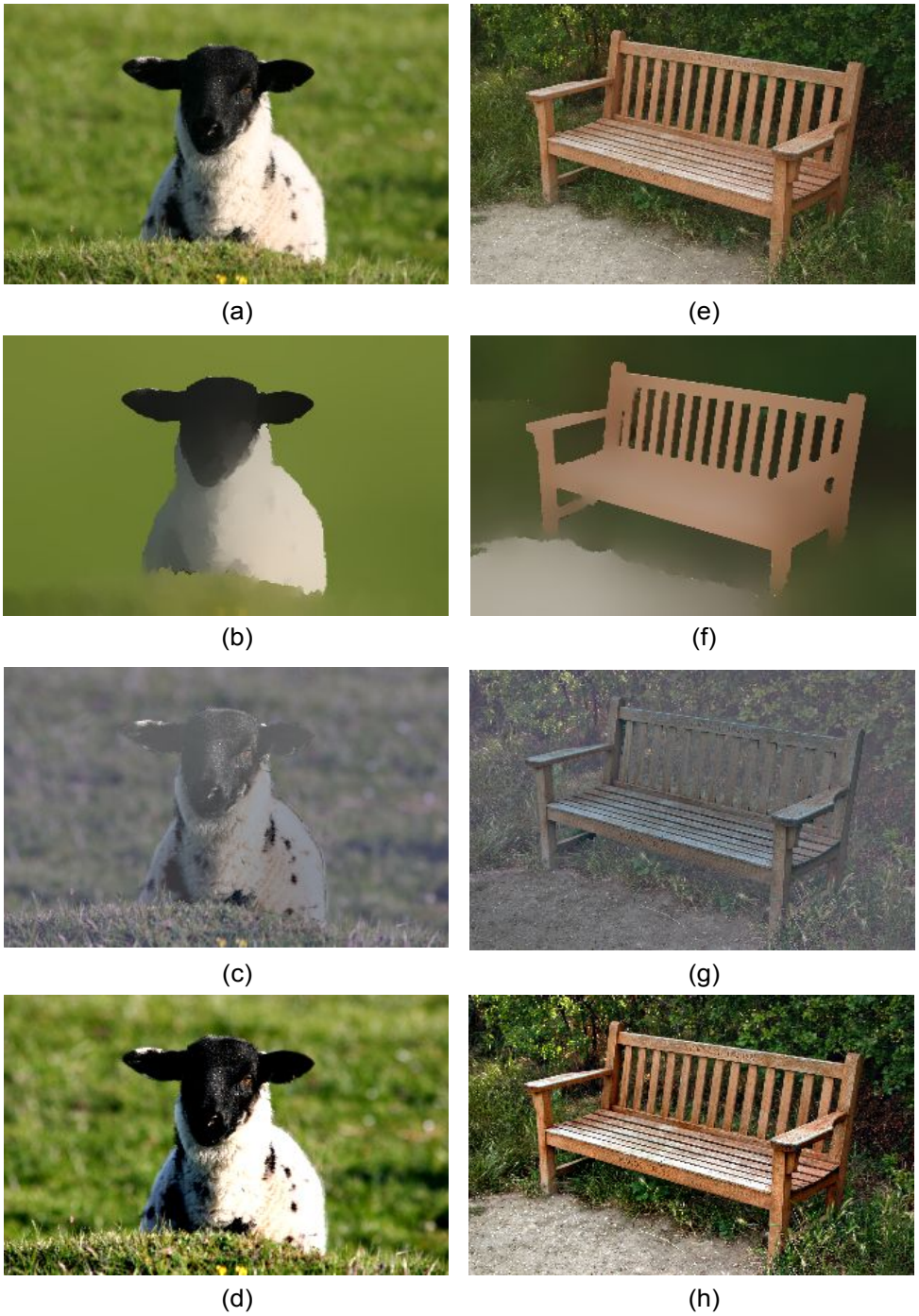

Figure 3.5: Detail Enhancement. (a) and (e) are the input images. (b) and (f) are our semantic image smoothing results. (c) and (g) are decomposed texture information outputs. (d) and (h) are the detail enhancement results. Best viewed in color.

detection approaches by applying them to our smoothed images Fig. 3.6 (b). Fig. 3.6 (d), (f), and (h) illustrate the refined edge detection results of Fig. 3.6 (c), (e), and (g) correspondingly.

3.4.3.3 Semantic Segmentation

In this section, we show that the smoothed image also can help semantic segmentation. Fully connected conditional random field [69], which is also called as dense conditional random field (Dense-CRF), is a very popular tool to refine semantic image segmentation results. We propose to use a modified version of Dense-CRF, which we call Dense-CRF+, where the smoothed images are used to model appearance kernel of Gaussian edge pairwise term instead of the typical RGB color vectors. For the sake of comparison with original Dense-CRF, we use the MSRC-21 dataset, the same data splits and unary potentials as the one used by [69].

Class	Unary	Dense-CRF	Dense-CRF+
Average	76.39	79.37	**79.55**
Overall	83.18	87.78	**88.01**

Table 3.1: The quantitative semantic segmentation results on the MSRC-21 dataset.

We choose two standard measures of multi-class segmentation accuracy as [69] used, *i.e.* Overall and Average. Overall is the pixel-wise labeling accuracy, which is computed over the whole image pixels for all classes. Average is the pixel-wise labeling accuracy computed for all classes and the averaged over these classes. The original ground truth labelings of the MSRC-21 dataset are relatively imprecise. There are some regions around objects boundaries left unlabeled. This makes it difficult to evaluate the quantitative performance of semantic segmentation results. Therefore, we evaluated our results on the 94 accurate ground truth labelings provided by [69], which is fully annotated at the pixel-level, with accurate labeling around complex boundaries. Table 3.1 shows the quantitative experimental results. We get the Average accuracy 79.55 and Overall accuracy 88.01. Our method outperforms the original Dense-CRF approach [69] on the MSRC-21 dataset. Fig. 3.7 shows some qualitative semantic

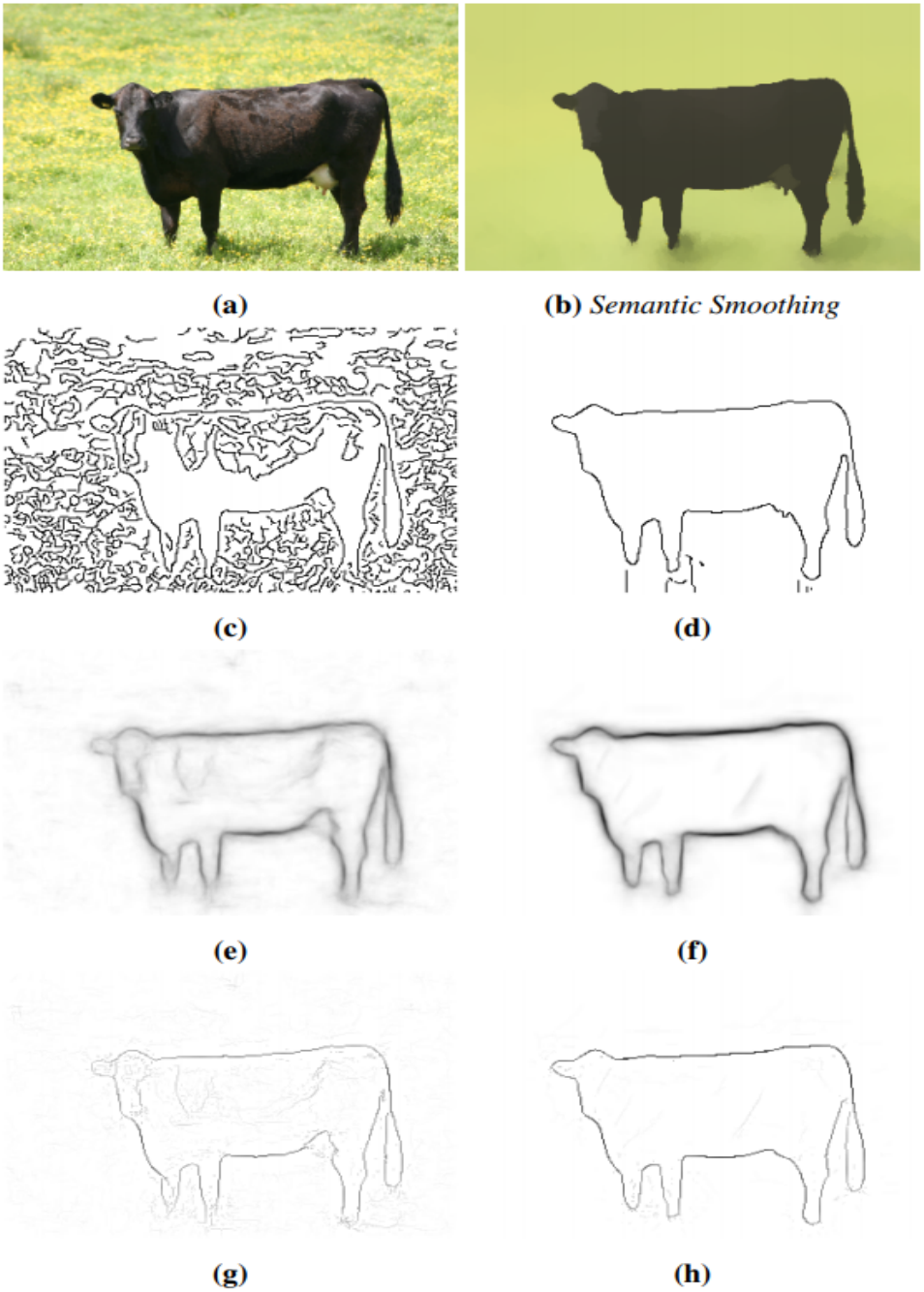

Figure 3.6: Edge Detection. (a) Input image, (c) Canny edge detection [15] applied to (a), (e) Structure edge detection [29] applied to (a), (g) Non-maximal suppression applied to (e). (b) Our semantic smoothing result, (d) Canny edge detection [15] applied to (b), (f) Structure edge detection [29] applied to (b), (h) Non-maximal suppression applied to (f).

segmentation results on the MSRC-21 dataset. Our Dense-CRF+ obtains more accurate results than the Dense-CRF, which produces many spatially disjoint object segments. As a future work, it is possible to jointly inference semantic smoothing and segmentation.

3.5 Conclusion

In this chapter, we propose a semantic-aware image smoothing method. Unlike previous image smoothing techniques which use the low-level vision features, such as appearance and gradient, or the mid-level vision features, such as edge or boundary detection, our proposed technique is developed based on the high-level semantic information of the image. Besides exploiting the high-level semantic information, our method also combines the low-level and the mid-level features. Effectiveness of our approach is demonstrated in different applications, including texture removal, detail enhancement, edge detection, and semantic segmentation. The limitation of the semantic smoothing is that it depends on the quality of the semantic segmentation. But with the development of semantic segmentation techniques, particularly using deep learning, we will have enough confidence to believe that using semantic information will be advantageous for image smoothing. In future work, we would like to extend our method by exploiting diverse levels of semantic information, such as instance segmentation [24], object part segmentation [134], and material segmentation [8].

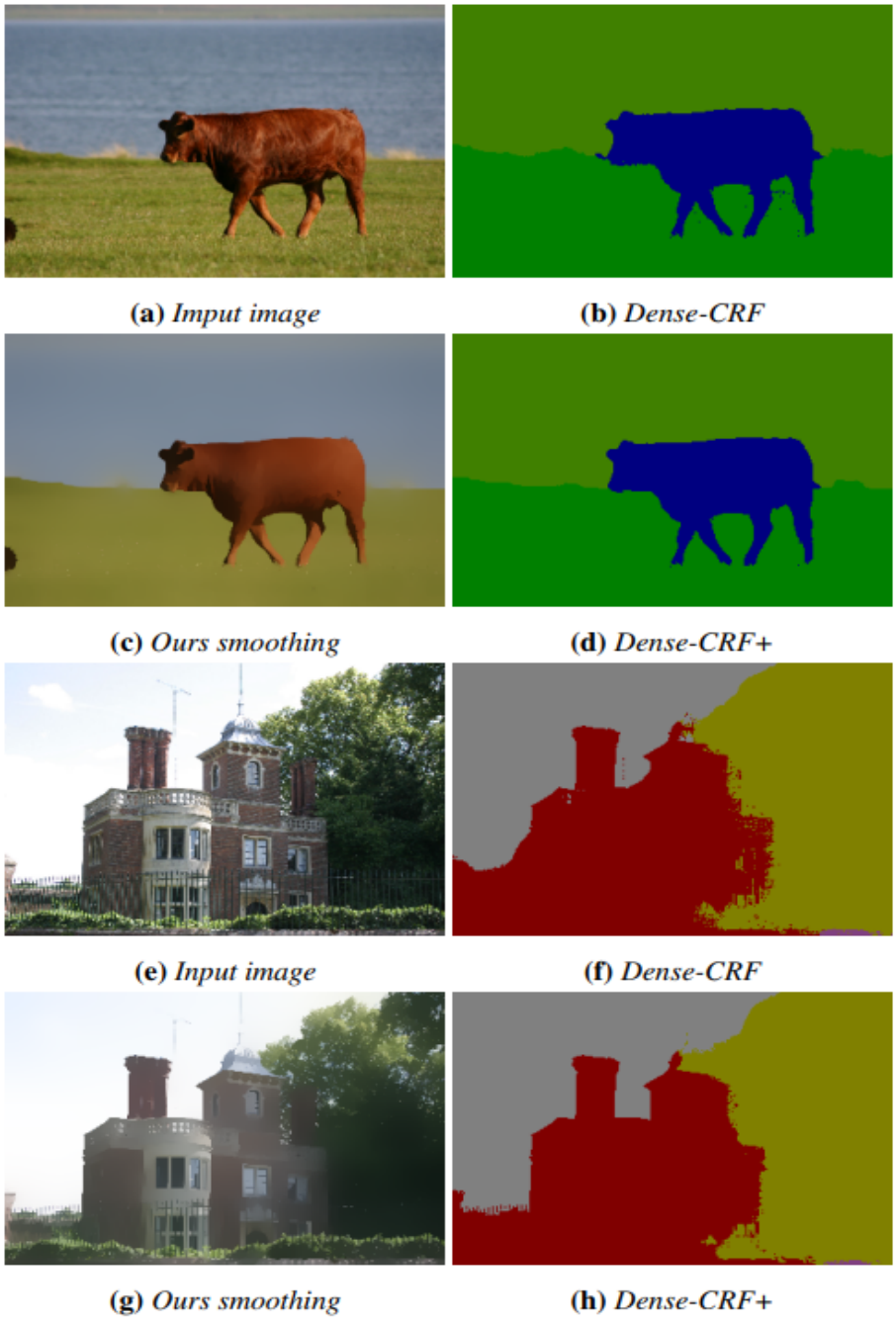

Figure 3.7: Semantic segmentation. (a) and (e) are input images. (b) and (f) are Dense-CRF segmentation results. (c) and (g) are our semantic smoothing results. (d) and (h) are Dense-CRF+ segmentation results. Our method predicts segmentations which are localized around object boundaries and are spatially smooth. Best viewed in color.

Chapter 4

Common Object Segmentation

In this chapter, we present a deep object co-segmentation (DOCS) approach for segmenting common objects of the same class within a pair of images. This means that the method learns to ignore common, or uncommon, background *stuff* and focuses on common *objects*. If multiple object classes are presented in the image pair, they are jointly extracted as foreground. To address this task, we propose a CNN-based Siamese encoder-decoder architecture. The encoder extracts high-level semantic features of the foreground objects, a mutual correlation layer detects the common objects, and finally, the decoder generates the output foreground masks for each image. To train our model, we compile a large object co-segmentation dataset consisting of image pairs from the PASCAL dataset with common objects masks. We evaluate our approach on commonly used datasets for co-segmentation tasks and observe that our approach consistently outperforms competing methods, for both seen and unseen object classes.

4.1 Introduction

Object co-segmentation is the task of segmenting the common objects from a set of images. It is applied in various computer vision applications and beyond, such as browsing in photo collections [104], 3D reconstruction [68], semantic segmentation [114], object-based image retrieval [129], video object tracking [104], and interactive image segmentation [104].

There are different challenges for object co-segmentation with varying level of difficulty: (1) Rother *et al.* [104] first proposed the term of *co-segmentation* as the task of segmenting the *common parts* of an image pair simultaneously. They showed that segmenting two images jointly achieves better accuracy in contrast to segmenting them independently. They assume that the common parts have a similar appearance. However, the background in both images are significantly different, see Fig. 4.1 (a). (2) Another challenge is to segment the same object instance or similar objects of the same class with low intra-class variation, even with similar background [7, 129], see Fig. 4.1 (b). (3) A more challenging task is to segment common objects from the same class with large variability in terms of scale, appearance, pose, viewpoint and background [105], see Fig. 4.1 (c).

All of the mentioned challenges assume that the image set contains only one common object and the common object should be salient within each image. In this work, we address a more general problem of co-segmentation without this assumption, *i.e.* multiple object classes can be presented within the images, see Fig. 4.1 (d). As it is shown, the co-segmentation result for one specific image including multiple objects can be different when we pair it with different images.Additionally, we are interested in co-segmenting objects, *i.e. things* rather than *stuff*. The idea of object co-segmentation was introduced by Vicente *et al.* [129] to emphasize the resulting segmentation to be a *thing* such as a 'cat' or a 'monitor', which excludes common, or uncommon, *stuff* classes like 'sky' or 'sea'.

Segmenting objects in an image is one of the fundamental tasks in computer vision. While image segmentation has received great attention during the recent rise of deep learning [89, 102, 152, 140, 100], the related task of object co-segmentation remains largely unexplored by newly developed deep learning techniques. Most of the recently proposed object co-segmentation methods still rely on models without feature learning. This includes methods utilizing superpixels, or proposal segments [129, 123] to extract a set of object candidates, or methods which use a complex CRF model [80, 100] with hand-crafted features [100] to find the segments with the highest similarity.

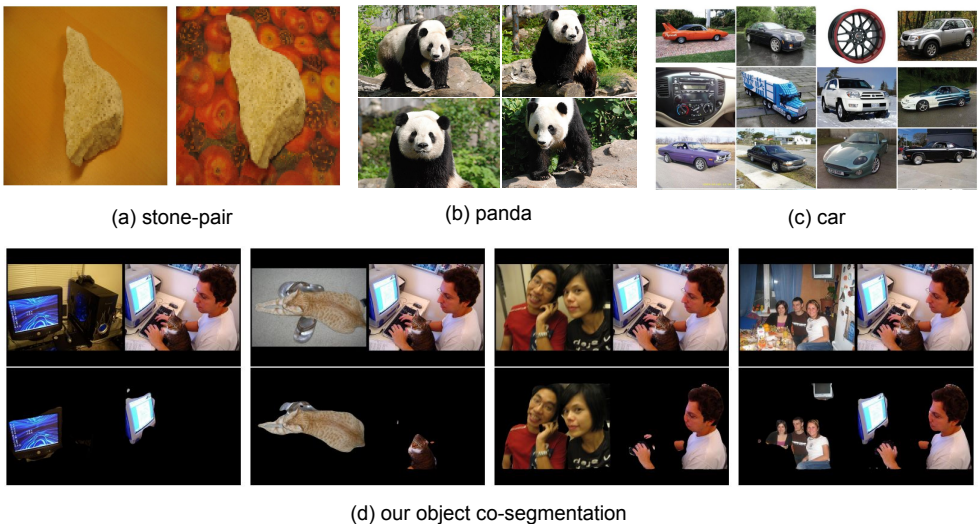

Figure 4.1: Different co-segmentation challenges: (a) segmenting common parts, in terms of small appearance deviation, with varying background [104], (b) segmenting common objects from the same class with low intra-class variation but similar background [7, 128], (c) segmenting common objects from the same class with large variability in terms of scale, appearance, pose, viewpoint and background [105]. (d) segmenting common objects in images including more than one object from multiple classes. Second row shows our predicted co-segmentation of these challenging images.

In this chapter, we propose a simple yet powerful method for segmenting objects of a common semantic class from a pair of images using a convolutional encoder-decoder neural network. Our method uses a pair of Siamese encoder networks to extract semantic features for each image. The mutual correlation layer at the network's bottleneck computes localized correlations between the semantic features of the two images to highlight the heat-maps of common objects. Finally, the Siamese decoder networks combine the semantic features from each image with the correlation features to produce detailed segmentation masks through a series of deconvolutional layers. Our approach is trainable in an end-to-end manner and does not require any, potentially long runtime, CRF optimization procedure at evaluation time. We perform an extensive evaluation of our deep object co-segmentation and show that our model can achieve state-of-the-art performance on multiple common co-segmentation datasets.

In summary, our main contributions are as follows:

- We propose a simple yet effective convolutional neural network (CNN) architecture for object co-segmentation that can be trained end-to-end. To the best of our knowledge, this is the first pure CNN framework for object co-segmentation, which does not depend on any hand-crafted features.
- We achieve state-of-the-art results on multiple object co-segmentation datasets, and introduce a challenging object co-segmentation dataset by adapting Pascal dataset for training and testing object co-segmentation models.

4.2 Related Work

We start by discussing object co-segmentation by roughly categorizing them into three branches: co-segmentation without explicit learning, co-segmentation with learning, and interactive co-segmentation. After that, we briefly discuss various image segmentation tasks and corresponding approaches based on CNNs.

4.2.1 Co-Segmentation without Explicit Learning

Rother *et al.* [104] proposed the problem of image co-segmentation for image pairs. They minimize an energy function that combines an MRF smoothness prior term with a histogram matching term. This forces the histogram statistic of common foreground regions to be similar. In a follow-up work, Mukherjee *et al.* [94] replace the l_1 norm in the cost function by an l_2 norm. In [48], Hochbaum and Singh used a reward model, in contrast to the penalty strategy of [104].In [128], Vicente *et al.* studied various models and showed that a simple model based on Boykov-Jolly [10] works the best. Joulin *et al.* [58] formulated the co-segmentation problem in terms of a discriminative clustering task. Rubio *et al.* [106] proposed to match regions, which results from an over-segmentation algorithm, to establish correspondences between the common objects. Rubinstein *et al.* [105] combined a visual saliency and dense correspondences, using SIFT flow, to capture the sparsity and visual variability of the common object

in a group of images. Fu *et al.* [38] formulated object co-segmentation for RGB-D input images as a fully-connected graph structure, together with mutex constraints. In contrast to these works, our method is a pure learning based approach.

4.2.2 Co-Segmentation with Learning

In [129], Vicente *et al.* generated a pool of object-like proposal-segmentations using constrained parametric min-cut [16]. Then they trained a random forest classifier to score the similarity of a pair of segmentation proposals. Yuan *et al.* [145] introduced a deep dense conditional random field framework for object co-segmentation by inferring co-occurrence maps. These co-occurrence maps measure the objectness scores, as well as, similarity evidence for object proposals, which are generated using selective search [127]. Similar to the constrained parametric min-cut, selective search also uses hand-crafted SIFT and HOG features to generate object proposals. Therefore, the model of [145] cannot be trained end-to-end. In addition, [145] assume that there is a single common object in a given image set, which limits application in real-world scenarios. Recently, Quan *et al.* [100] proposed a manifold ranking algorithm for object co-segmentation by combining low-level appearance features and high-level semantic features. However, their semantic features are pre-trained on the ImageNet dataset. In contrast, our method is based on a pure CNN architecture, which is free of any hand-crafted features and object proposals and does not depend on any assumption about the existence of common objects.

4.2.3 Interactive Co-Segmentation

Batra *et al.* [7] firstly presented an algorithm for interactive co-segmentation of a foreground object from a group of related images. They use users' scribbles to indicate the foreground. Collins *et al.* [21] used a random walker model to add consistency constraints between foreground regions within the interactive co-segmentation framework. However, their co-segmentation results are sensitive to the size and positions of users' scribbles. Dong *et al.* [30] proposed

an interactive co-segmentation method which uses global and local energy optimization, whereby the energy function is based on scribbles, inter-image consistency, and a standard local smoothness prior. In contrast, our work is not a user-interactive co-segmentation approach.

4.2.4 CNNs for Image Segmentation

In the last few years, CNNs have achieved great success for the tasks of image segmentation, such as semantic segmentation [89, 95, 144, 87, 140, 150], interactive segmentation [140, 139], and salient object segmentation [81, 133, 51].

Semantic segmentation aims at assigning semantic labels to each pixel in an image. Fully convolutional networks (FCN) [89] became one of the first popular architectures for semantic segmentation. Nor *et al.* [95] proposed a deep deconvolutional network to learn the upsampling of low-resolution features. Both U-Net [102] and SegNet [4] proposed an encoder-decoder architecture, in which the decoder network consists of a hierarchy of decoders, each corresponding to an encoder. Yu *et al.* [144] and Chen *et al.* [17] proposed dilated convolutions to aggregate multi-scale contextual information, by considering larger receptive fields. Salient object segmentation aims at detecting and segmenting the salient objects in a given image. Recently, deep learning architectures have become popular for salient object segmentation [81, 133, 51]. Li and Yu [81] addressed salient object segmentation using a deep network which consists of a pixel-level multi-scale FCN and a segment scale spatial pooling stream. Wang *et al.* [133] proposed recurrent FCN to incorporate saliency prior knowledge for improved inference, utilizing a pre-training strategy based on semantic segmentation data. Jain *et al.* [51] proposed to train an FCN to produce pixel-level masks of all "object-like" regions given a single input image.

Although CNNs play a central role in image segmentation tasks, there has been no prior work with a pure CNN architecture for object co-segmentation. To the best of our knowledge, our deep CNN architecture is the first of its kind for object co-segmentation.

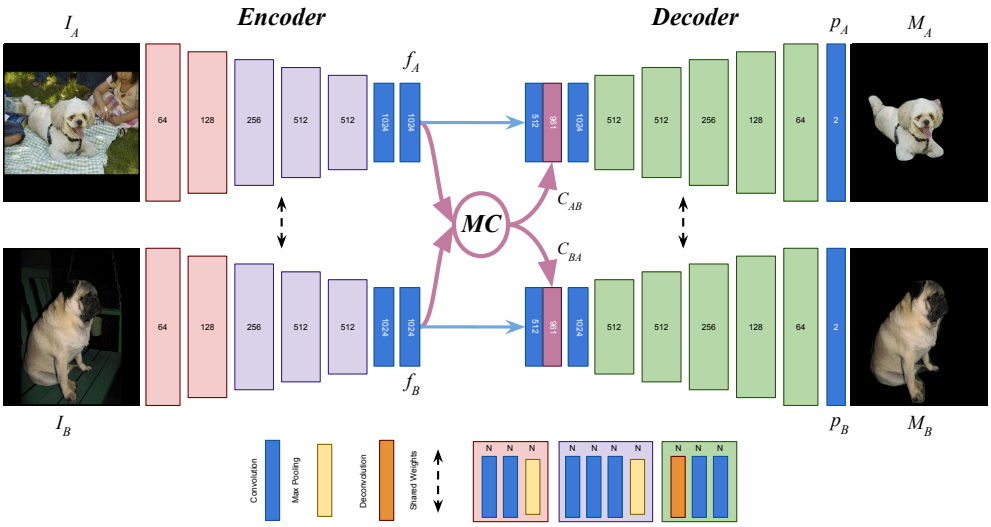

Figure 4.2: Deep Object Co-Segmentation. Our network includes three parts: (i) passing input images I_A and I_B through a Siamese encoder to extract feature maps f_A and f_B, (ii) using a mutual correlation network to perform feature matching to obtain correspondence maps C_{AB} and C_{BA}, (iii) passing concatenation of squeezed feature maps and correspondence maps through a Siamese decoder to get the common objects masks M_A and M_B.

4.3 Method

In this section, we introduce a new CNN architecture for segmenting the common objects from two input images. The architecture is end-to-end trainable for the object co-segmentation task. Fig. 4.2 illustrates the overall structure of our architecture. Our network consists of three main parts: (1) Given two input images I_A and I_B, we use a Siamese encoder to extract high-level semantic feature maps f_A and f_B. (2) Then, we propose a mutual correlation layer to obtain correspondence maps C_{AB} and C_{BA} by matching feature maps f_A and f_B at pixel-level. (3) Finally, given the concatenation of the feature maps f_A and f_B and correspondence maps C_{AB} and C_{BA}, a Siamese decoder is used to obtain and refine the common object masks M_A and M_B.

In the following, we first describe each of the three parts of our architecture in detail. Then in Sec 4.3.4, the loss function is introduced. Finally, in Sec 4.3.5,

we explain how to extend our approach to handle co-segmentation of a group of images, *i.e.* going beyond two images.

4.3.1 Siamese Encoder

The first part of our architecture is a Siamese encoder which consists of two identical feature extraction CNNs with shared parameters. We pass the input image pair I_A and I_B through the Siamese encoder network pair to extract feature maps f_A and f_B. More specifically, our encoder is based on the VGG16 network [118]. We keep the first 13 convolutional layers and replace *fc6* and *fc7* with two 3×3 convolutional layers *conv6-1* and *conv6-2* to produce feature maps which contain more spatial information. In total, our encoder network has 15 convolutional layers and 5 pooling layers to create a set of high-level semantic features f_A and f_B. The input to the Siamese encoder is two 512×512 images and the output of the encoder is two 1024-channel feature maps with a spatial size of 16×16.

4.3.2 Mutual Correlation

The second part of our architecture is a mutual correlation layer. The outputs of encoders f_A and f_B represent the high-level semantic content of the input images. When the two images contain objects that belong to a common class, they should contain similar features at the locations of the shared objects. Therefore, we propose a mutual correlation layer to compute the correlation between each pair of locations on the feature maps. The idea of utilizing the correlation layer is inspired by Flownet [31], in which the correlation layer is used to match feature points between frames for optical flow estimation. Our motivation of using the correlation layer is to filter the heat-maps (high-level features), which are generated separately for each input image, to highlight the heat-maps on the common objects (see Fig. 4.3).

In detail, the mutual correlation layer performs a pixel-wise comparison between two feature maps f_A and f_B. Given a point (i, j) and a point (m, n) inside a patch around (i, j), the correlation between feature vectors $f_A(i, j)$ and

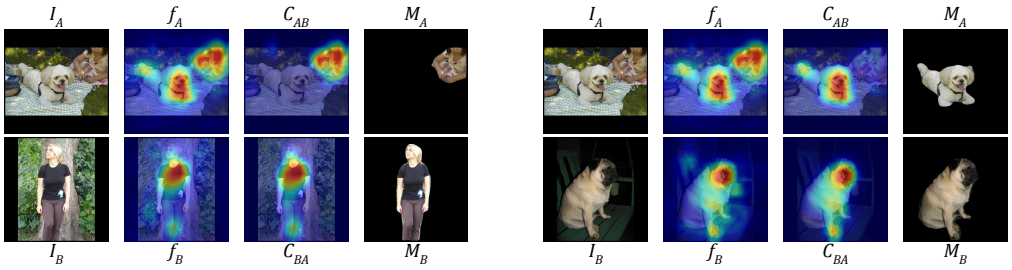

Figure 4.3: The visualization of the heat-maps. Given a pair of input images I_A and I_B, after passing them through the Siamese encoder, we extract feature maps f_A and f_B. We use the mutual correlation layer to perform feature matching to obtain correspondence maps C_{AB} and C_{BA}. Then, using our Siamese decoder we predict the common objects masks M_A and M_B. As shown before correlation layer, the heat-maps are covering all the objects inside the images. After applying the correlation layer, the heat-maps on uncommon objects are filtered out. Therefore, we utilize the output of the correlation layer to guide the network for segmenting the common objects.

$f_B(m, n)$ is defined as

$$C_{AB}(i, j, k) = \langle\, f_A(i, j), f_B(m, n) \rangle \tag{4.1}$$

where $k = (n - j)D + (m - i)$ and $D \times D$ is patch size. Since the common objects can locate at any place on the two input images, we set the patch size to $D = 2 * max(w - 1, h - 1) + 1$, where w and h are the width and height of the feature maps f_A and f_B. The output of the correlation layer is a feature map C_{AB} of size $w \times\ h \times D^2$. We use the same method to compute the correlation map C_{BA} between f_B and f_A.

4.3.3 Siamese Decoder

The Siamese decoder is the third part of our architecture, which predicts two foreground masks of the common objects. We squeeze the feature maps f_A and f_B and concatenate them with their correspondence maps C_{AB} and C_{BA} as the input to the Siamese decoder (Fig. 4.2). The same as the Siamese encoder, the decoder is also arranged in a Siamese structure with shared parameters. There are five blocks in our decoder, whereby each block has one deconvolutional layer and two convolutional layers. All the convolutional and deconvolutional

layers in our Siamese decoder are followed by a ReLU activation function. By applying a Softmax function, the decoder produces two probability maps p_A and p_B. Each probability map has two channels, background and foreground, with the same size as the input images.

4.3.4 Loss Function

We define our object co-segmentation as a binary image labeling problem and use the standard cross entropy loss function to train our network. The full loss score $\mathcal{L}_{AB}$ is then estimated by

$$\mathcal{L}_{AB} = \mathcal{L}_A + \mathcal{L}_B, \tag{4.2}$$

where the $\mathcal{L}_A$ and the $\mathcal{L}_B$ are cross-entropy loss functions for the image A and the image B, respectively.

4.3.5 Group Co-Segmentation

Although our architecture is trained for image pairs, our method can handle a group of images. Given a set of N images $\mathcal{I} = \{I_1, ..., I_N\}$, we pair each image with $K \leq N - 1$ other images from $\mathcal{I}$. Then, we use our DOCS network to predict the probability maps for the pairs, $\mathcal{P} = \left\{p_n^k : 1 \leq n \leq N, 1 \leq k \leq K\right\}$, where p_n^k is the predicted probability map for the kth pair of image I_n. Finally, we compute the final mask M_n for image I_n as

$$M_n(x, y) = \text{median}\{p_n^k(x, y)\} > \sigma. \tag{4.3}$$

where σ is the acceptance threshold. In this work, we set $\sigma = 0.5$. We use the median to make our approach more robust to groups with outliers.

4.4 Experiments

4.4.1 Datasets

Training a CNN requires a lot of data. However, existing co-segmentation datasets are either too small or have a limited number of object classes. The

MSRC dataset [117] was first introduced for supervised semantic segmentation, then a subset was used for object co-segmentation [129]. This subset of MSRC only has 7 groups of images and each group has 10 images. The iCoseg dataset, introduced in [7], consists of several groups of images and is widely used to evaluate co-segmentation methods. However, each group contains images of the same object instance or very similar objects from the same class. The Internet dataset [105] contains thousands of images obtained from the Internet using image retrieval techniques. However, it only has three object classes: *car, horse* and *airplane*, where images of each class are mixed with other noise objects. In [34], Faktor and Irani use PASCAL dataset for object co-segmentation. They separate the images into 20 groups according to the object classes and assume that each group only has one object. However, this assumption is not common for natural images.

Inspired by [34], we create an object co-segmentation dataset by adapting the PASCAL dataset labeled by [43]. The original dataset consists of 20 foreground object classes and one background class. It contains $8,498$ training and $2,857$ validation pixel-level labeled images. From the training images, we sampled $161,229$ pairs of images, which have common objects, as a new co-segmentation training set. We used PASCAL validation images to sample $42,831$ validation pairs and $40,303$ test pairs. Since our goal is to segment the common objects from the pair of images, we discard the object class labels and instead we label the common objects as foreground. Fig. 4.1(d) shows some examples of image pairs of our object co-segmentation dataset. In contrast to [34], our dataset consists of image pairs of one or more arbitrary common classes.

4.4.2 Implementation Details and Runtime

We use the Caffe framework [55] to design and train our network. We use our co-segmentation dataset for training. We did not use any images from the MSRC, Internet or iCoseg datasets to fine tune our model. The *conv1-conv5* layers of our Siamese encoder (VGG-16 net [118]) are initialized with weights trained on the Imagenet dataset [26]. We train our network on one GPU for 100K iterations using Adam solver [62]. We use small mini-batches of 10 image

pairs, a momentum of 0.9, a learning rate of $1e-5$, and a weight decay of 0.0005.

Our method can handle a large set of images in linear time complexity $\mathcal{O}(N)$. As mentioned in Sec. 4.3.5 in order to co-segment an image, we pair it with K ($K \leq N-1$) other images. In our experiments, we used all possible pairs to make the evaluations comparable to other approaches. Although in this case our time complexity is quadratic $\mathcal{O}(N^2)$, our method is significantly faster than others.

Number of images	Others time	Our time
2	8 minutes [58]	0.1 seconds
30	4 to 9 hours [58]	43.5 seconds
30	22.5 minutes [132]	43.5 seconds
418 (14 classes, $\sim$ 30 images per class)	29.2 hours [34]	10.15 minutes
418 (14 classes, $\sim$ 30 images per class)	8.5 hours [54]	10.15 minutes

Table 4.1: The computation time comparison between the different methods.

To show the influence of number of pairs K, we validate our method on the Internet dataset *w.r.t.* K (as shown in Table 4.2). Each image is paired with K random images from the set. As shown, we achieve state-of-the-art performance even with $K = 10$. Therefore, the complexity of our approach is $\mathcal{O}(KN) = \mathcal{O}(N)$ which is linear with respect to the group size.

Internet	K=10		K=20		K=99(all)	
(N=100)	Precision	Jaccard	Precision	Jaccard	Precision	Jaccard
Car	93.93	82.89	93.91	82.85	93.90	82.81
Horse	92.31	69.12	92.35	69.17	92.45	69.44
Airplane	94.10	65.37	94.12	65.45	94.11	65.43
Average	93.45	72.46	93.46	72.49	93.49	72.56

Table 4.2: Influence of number of pairs K.

4.4.3 Results

We report the performance of our approach on MSRC [117, 128], Internet [105], and iCoseg [7] datasets, as well as our own co-segmentation dataset.

4.4.3.1 Metrics.

For evaluating the co-segmentation performance, there are two common metrics. The first one is *Precision*, which is the percentage of correctly segmented pixels of both foreground and background masks. The second one is *Jaccard*, which is the intersection over union of the co-segmentation result and the ground truth foreground segmentation.

4.4.3.2 PASCAL Co-Segmentation.

As we mentioned in Sec 4.4.1, our object co-segmentation dataset consists of $40,303$ test image pairs. We evaluate the performance of our method on our co-segmentation test data. We also tried to obtain the common objects of same classes using a deep semantic segmentation model, here FCN8s [89]. First, we train FCN8s with the PASCAL dataset. Then, we obtain the common objects from two images by predicting the semantic labels using FCN8s and keeping the segments with common classes as foreground. Our co-segmentation method (**94.2%** for *Precision* and **64.5%** for *Jaccard*) outperforms FCN8s (**93.2%** for *Precision* and **55.2%** for *Jaccard*), which uses the same VGG encoder, and trained with the same training images. The improvement is probably due to the fact that our DOCS architecture is specifically designed for the object co-segmentation task, which FCN8s is designed for the semantic labeling problem. Another potential reason is that generating image pairs is a form of data augmentation. We would like to exploit these ideas in the future work. Fig. 4.4 shows the qualitative results of our approach on the PASCAL co-segmentation dataset. We can see that our method successfully extracts different foreground objects for the left image when paired with a different image to the right.

4.4.3.3 MSRC.

The MSRC subset has been used to evaluate the object co-segmentation performance by many previous methods [128, 105, 34, 132]. For the fair comparison, we use the same subset as [128]. We use our group co-segmentation method to extract the foreground masks for each group. In Table. 4.3, we show the quantitative results of our method as well as four state-of-the-art methods

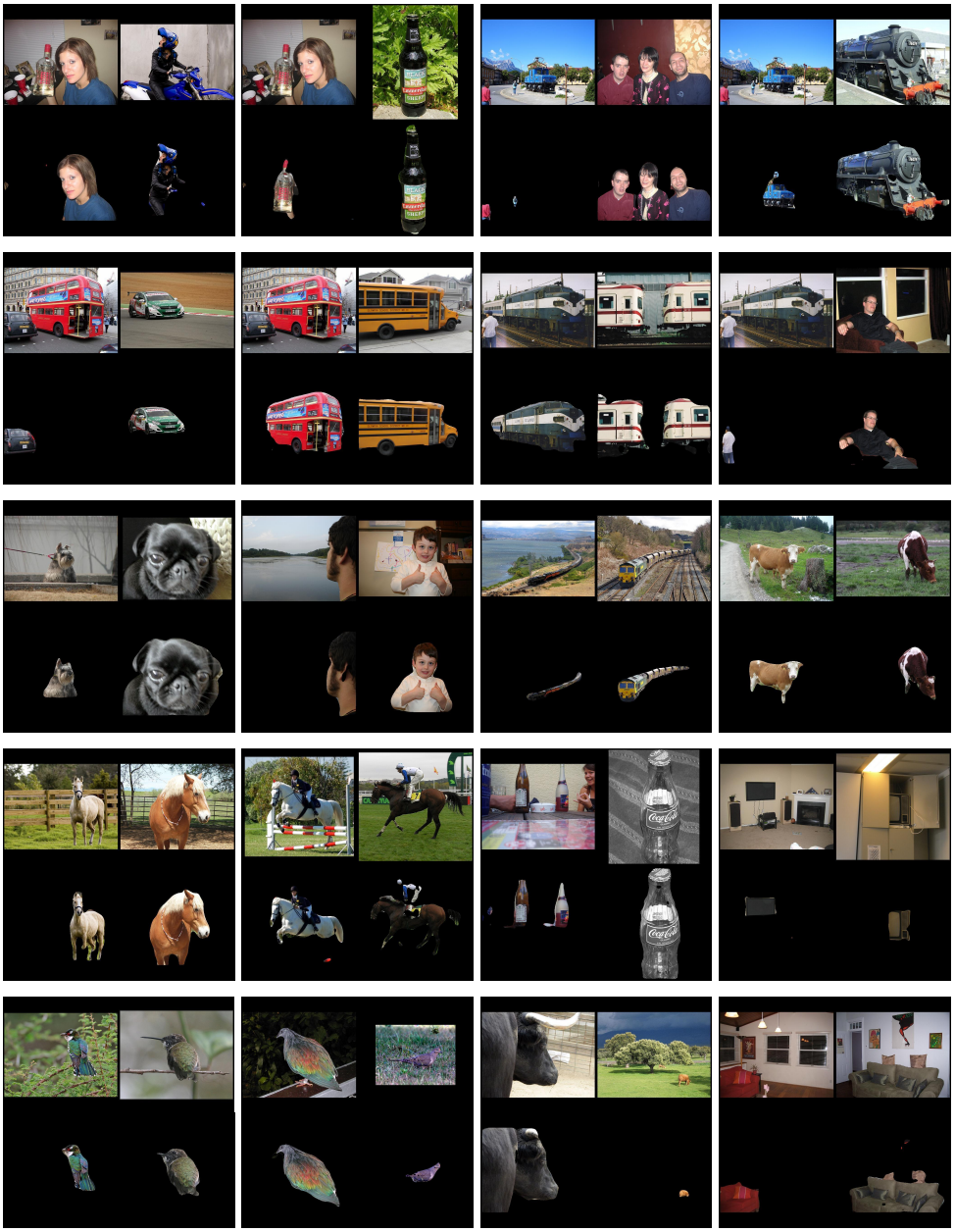

Figure 4.4: Our qualitative results on PASCAL Co-segmentation dataset. (odd rows) the input images, (even rows) the corresponding objec co-segmentation results.

MSRC	[129]	[105]	[132]	[34]	Ours
Precision	90.2	92.2	92.2	92.0	**95.4**
Jaccard	70.6	74.7	-	77.0	**82.9**

Table 4.3: Quantitative results on the MSRC dataset (seen classes). Quantitative comparison results of our DOCS approach with four state-of-the-art co-segmentation methods on the co-segmentation subset of the MSRC dataset.

[129, 105, 34, 132]. Our *Precision* and *Jaccard* show a significant improvement compared to previous methods. It is important to note that [129] and [132] are supervised methods, i.e. both use images of the MSRC dataset to train their models. We obtain the new state-of-the-art results on this dataset even without training or fine-tuning on any images from the MSRC dataset. Visual examples of object co-segmentation results on the subset of the MSRC dataset can be found in Fig. 4.5.

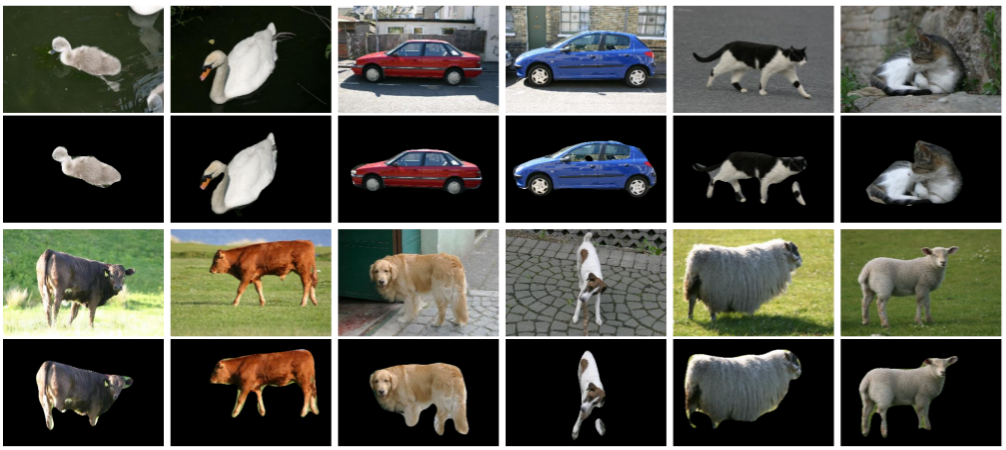

Figure 4.5: Our qualitative results on the MSRC dataset (seen classes). (odd rows) the input images, (even rows) the corresponding object co-segmentation results.

4.4.3.4 Internet.

In our experiment, for the fair comparison, we followed [105, 18, 100, 145] to use the subset of the Internet dataset to evaluate our method. In this subset, there are 100 images in each category. We compare our method with five previous

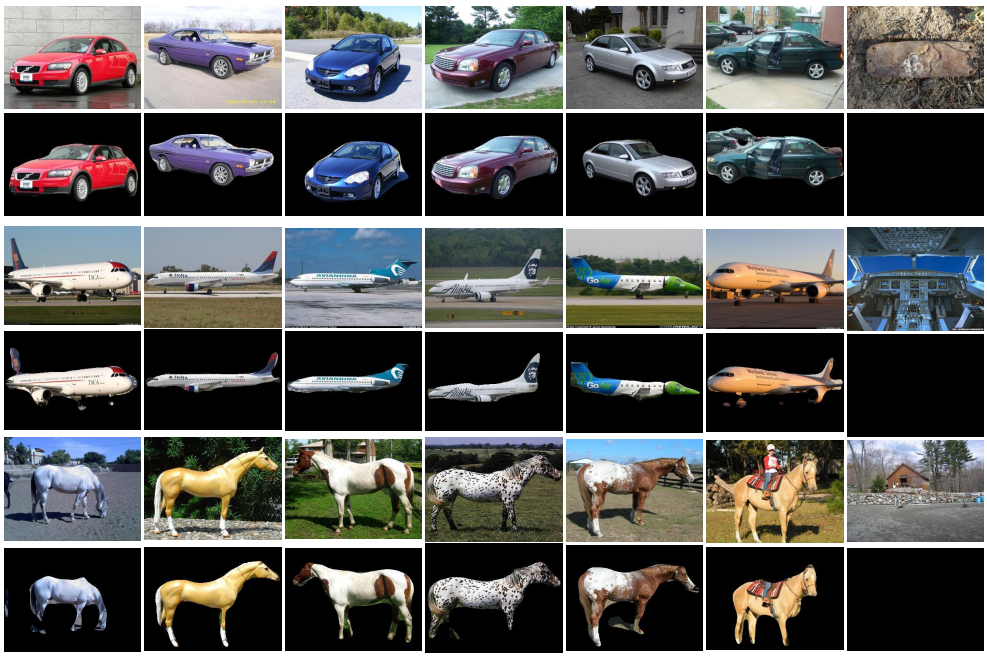

Figure 4.6: Our qualitative results on the Internet dataset (seen classes). (odd rows) the input images, (even rows) the corresponding object co-segmentation results.

Internet		[58]	[105]	[18]	[100]	[145]	Ours
Car	P	58.7	85.3	87.6	88.5	90.4	**93.9**
	J	37.1	64.4	64.9	66.8	72.0	**82.8**
Horse	P	63.8	82.8	86.2	89.3	90.2	**92.4**
	J	30.1	51.6	33.4	58.1	65.0	**69.4**
Airplane	P	49.2	88.0	90.3	92.6	91.0	**94.1**
	J	15.3	55.8	40.3	56.3	**66.0**	65.4
Average	P	57.2	85.4	88.0	89.6	91.1	**93.5**
	J	27.5	57.3	46.2	60.4	67.7	**72.6**

Table 4.4: Quantitative results on the Internet dataset (seen classes). Quantitative comparison of our DOCS approach with several state-of-the-art co-segmentation methods on the co-segmentation subset of the Internet dataset. 'P' is the *Precision*, and 'J' is the *Jaccard*.

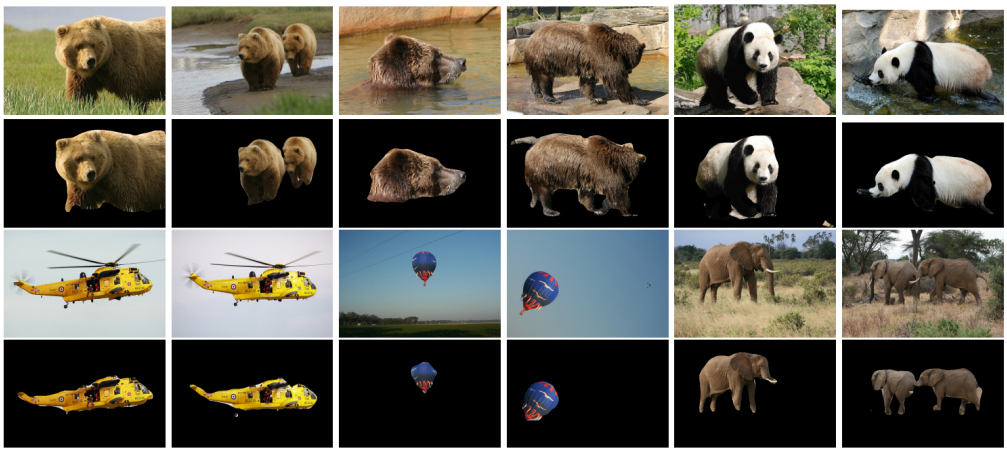

Figure 4.7: Our qualitative results on iCoseg dataset (unseen classes). Some results of our object co-segmentation method, with input image pairs in the odd rows and the corresponding object co-segmentation results in the even rows. For this dataset, the object classes were not known during training of our method (*i.e. unseen*).

approaches [58, 18, 105, 100, 145]. Table 4.4 shows the quantitative results of each object category with respect to *Precision* and *Jaccard*. We outperform most of the previous methods [58, 18, 105, 100, 145] in terms of *Precision* and *Jaccard*. Note that [145] is a supervised co-segmentation method, [18] trained a discriminative Latent-SVM detector and [100] used a CNN trained on the ImageNet to extract semantic features. Fig. 4.6 shows some quantitative results of our method. It can be seen that even for the 'noise' images in each group, our method can successfully recognize them. We show the 'noise' images in the last column.

4.4.3.5 iCoseg

To show that our method can generalize on *unseen classes*, *i.e.* classes which are not part of the training data, we need to evaluate our method on *unseen classes*. Batra *et al.* [7] introduced the iCoseg dataset for the *interactive* co-segmentation task. In contrast to the MSRC and Internet datasets, there are multiple object classes in the iCoseg dataset which do not appear in PASCAL VOC dataset. Therefore, it is possible to use the iCoseg dataset to evaluate the generalization

iCoseg	[105]	[53]	[34]	[54]	Ours
bear2	65.3	70.1	72.0	67.5	**88.7**
brownbear	73.6	66.2	**92.0**	72.5	91.5
cheetah	69.7	75.4	67.0	**78.0**	71.5
elephant	68.8	73.5	67.0	79.9	**85.1**
helicopter	80.3	76.6	**82.0**	80.0	73.1
hotballoon	65.7	76.3	88.0	80.2	**91.1**
panda1	75.9	80.6	70.0	72.2	**87.5**
panda2	62.5	71.8	55.0	61.4	**84.7**
average	70.2	73.8	78.2	74.0	**84.2**

Table 4.5: Quantitative results on the iCoseg dataset (unseen classes). Quantitative comparison of our DOCS approach with four state-of-the-art co-segmentation methods on some object classes of the iCoseg dataset, in terms of Jaccard. For this dataset, these object classes were not known during training of our method (*i.e. unseen*).

of our method on *unseen object classes*. We choose eight groups of images from the iCoseg dataset as our unseen object classes, which are *bear2, brown_bear, cheetah, elephant, helicopter, hotballoon, panda1* and *panda2*. There are two reasons for this choice: firstly, these object classes are not included in the PASCAL VOC dataset. Secondly, in order to focus on *objects*, in contrast to *stuff*, we ignore groups like *pyramid, stonehenge* and *taj-mahal*. We compare our method with four state-of-the-art approaches [53, 105, 34, 54] on unseen objects of the iCoseg dataset. Table 4.5 shows the comparison results of each unseen object groups in terms of *Jaccard*. The results show that for 5 out of 8 object groups our method performs best, and it is also superior on average. Note that the results of [53, 105, 34, 54] are taken from Table X in [54]. Fig. 4.7 shows some qualitative results of our method. It can be seen that our object co-segmentation method can detect and segment the common objects of these unseen classes accurately.

Furthermore to show the effect of number of PASCAL classes on the performance of our approach on unseen classes, we train our network on partial randomly picked PASCAL classes, *i.e.* $\{5, 10, 15\}$, and evaluate it on the iCoseg unseen classes. As it is shown in Table 4.6, our approach can generalize to

unseen classes even when it is trained with only 10 classes from PASCAL.

iCoseg	P(5)	P(10)	P(15)	P(20)
average	75.5	83.9	83.7	84.2

Table 4.6: Analyzing the effect of number of training classes on unseen classes.

4.4.4 Ablation Study

To show the impact of the mutual correlation layer in our network architecture, we design a baseline network *DOCS-Concat* without using mutual correlation layers. In detail, we removed the correlation layer and we concatenate f_A and f_B (instead of C_{AB}) for image I_A and concatenate f_B and f_A (instead of C_{BA}) for image I_B. In Table 4.7, we compare the performance of different network designs on multiple datasets. As shown, the mutual correlation layer in *DOCS-Corr* improved the performance significantly.

	DOCS-Concat		DOCS-Corr	
	Precision	Jaccard	Precision	Jaccard
Pascal VOC	92.6	49.9	**94.2**	**64.5**
MSRC	92.6	72.0	**95.4**	**82.9**
Internet	91.8	62.7	**93.5**	**72.6**
iCoseg(unseen)	93.6	78.9	**95.1**	**84.2**

Table 4.7: Impact of mutual correlation layer.

4.5 Conclusions

In this chapter, we present a new and efficient CNN-based method for solving the problem of object class co-segmentation, which consists of jointly detecting and segmenting objects belonging to a common semantic class from a pair of images. Based on a simple encoder-decoder architecture, combined with the mutual correlation layer for matching semantic features, we achieve state-of-the-art performance on various datasets, and demonstrate good generalization performance on segmenting objects of new semantic classes, unseen during

training. To train our model, we compile a large object co-segmentation dataset consisting of image pairs from PASCAL dataset with shared objects masks.

Chapter 5

Common Object Localization

We propose an approach to localize common objects from novel object categories in a set of images. We solve this problem using a new common component activation map in which we treat the class-specific activation maps as components to discover the common components in the image set. We show that our approach can generalize on novel object categories in our experiments.

5.1 Introduction

Learning to classify and localize visual objects is a fundamental problem in visual recognition. The task of object localization aims to recognize the category of the main object presents in the image and locate it with an axis-aligned bounding box [108]. Recently, most of the state-of-the-art object detection or localization methods [112, 101] are trained with a strong supervised manner, which requires a large amount of human labeled bounding box annotations. However, these annotations are expensive, particularly for the large-scale datasets, such as ImageNet [108].

Currently, there have been a lot of works solving object localization task using weakly supervised setting [96, 153, 148, 149], which learn object locations in a given image only using image-level category labels. Weakly supervised object localization is getting more attention since it does not need massive bonding box annotations for training. Zhou *et al.* [153] proposed Class Activation Maps (CAM) to generate *class-specific* localization maps using classification-trained

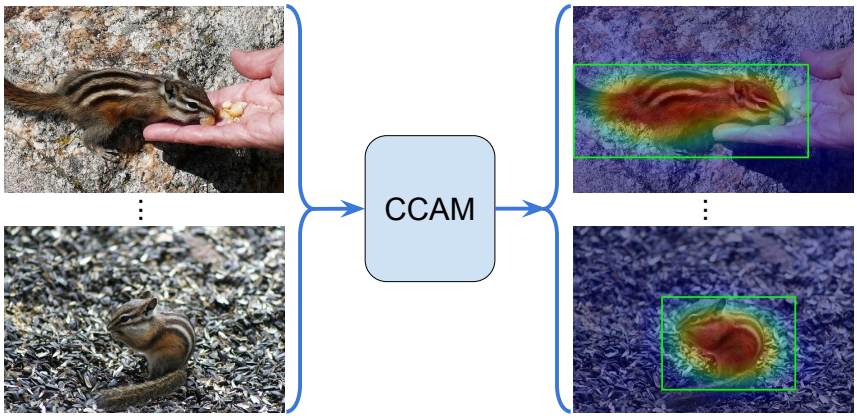

Figure 5.1: Common Object Localization. Given a set of images containing objects from novel classes (unseen during training), CCAM localizes the common objects in these images. Best viewed in colour

Convolutional Neural Networks (CNNs) with global average pooling. For a particular category, the class activation map shows the discriminative important image regions used by a CNN to recognize that category. However, these CAM related *class-specific* object localization methods [153, 111, 72, 148, 149] can only generate the localization maps of predefined object categories, which are not suitable for localizing the image regions for the unseen or unknown object classes.

Common object localization, also known as object co-localization, is the problem of localizing common objects of the same class across a set of distinct images [122, 59, 19, 86, 79, 82]. In contrast to weakly supervised object localization methods, the co-localization problem is not limited to predefined object categories.

In this chapter, we consider both weakly supervised object localization and object co-localization to propose a simple yet effective common object localization method for unseen object categories. Unlike previous works [122, 59, 19, 86], our approach is proposal-free, which does not need any object proposals to perform object localization and only requires a CNN model with similar architecture as [153], pre-trained on a classification task. We regard the output of the last fully-connected layer as a component vector for an input object, in-

stead of the categorical output for probability map. For a group of images, we first compute the average of the component vectors to find the group common vector. Then, we pick k largest entries from the group common vector. Finally, for each image, we compute a weighted sum of feature maps of the last convolutional layer to get the common component activation map according to the top k components. We test our method on six unseen ImageNet classes [86], which are not included in the 1000 categories used for training the CNN classification model. We show the effectiveness of our method in the result section.

5.2 Method

For making the paper self-contained, we first briefly review the class activation map (CAM) for the class-specific heatmap generation, then we show how to generalize the CAM to common component activation map (CCAM) to localize the common objects.

5.2.1 Class Activation Map

For a specific object category, the CAM indicates the discriminative image regions used by a CNN to identify the importance of that category. Given an input image I, we first pass it through a classification network [153], which uses global average pooling on the last convolutional layer and use those as features for a fully-connected layer to produce the object categorical output. Let F represent the feature maps of the last fully convolutional layer. The size of F is $H \times W \times C$, where $H \times W$ is the spatial size and C is the number of feature channels. We denote the weight matrix of the fully-connected layer as W, in which W_c^s is the weight corresponding to class s for the channel c and indicates the importance of the channel c for the specific class s. Then, the class activation map for the class s is defined as

$$M_s(h, w) = \sum_c W_c^s F_c(h, w). \tag{5.1}$$

For the specific class s, $M_s(x, y)$ can directly show up the importance of the activation at the spatial grid (h, w). Fig. 5.2 illustrates the procedure for generating these maps.

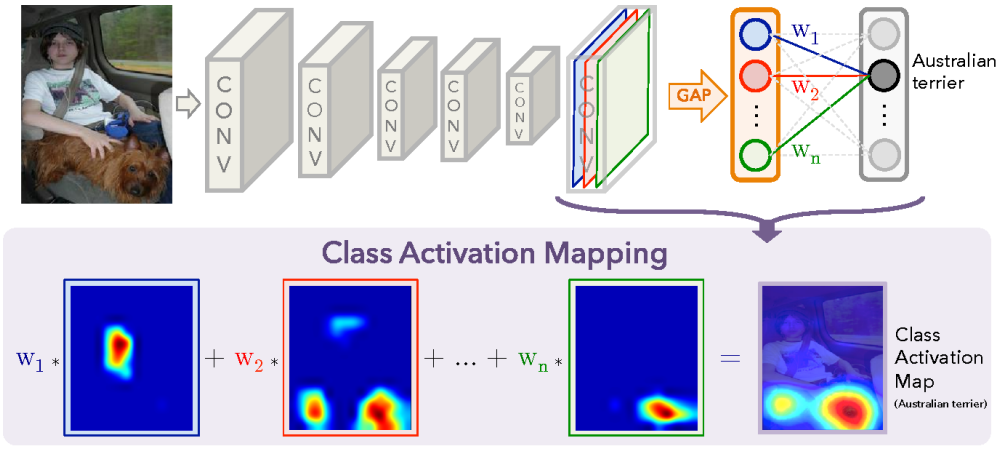

Figure 5.2: Class Activation Mapping. The figure is borrowed from [153].

5.2.2 Common Component Activation Map

For a given image with known categories, the CAM can identify the image regions which are most relevant to these particular categories. However, this method is incapable to find the important regions for the unseen object classes, which are not included in the training dataset. In order to generate the activation map for the unseen object, we treat the output of the fully connected layer as a component vector for the input image, instead of categorical probability maps. For a given group of N images $\mathcal{I} = \{I_1, ..., I_N\}$ containing objects from an unseen category, let the vectors $\mathcal{V} = \{V_1, ..., V_N\}$ be the outputs of the softmax function. Then we obtain the common component of the group by computing the average of output vectors $\mathcal{V}$ as

$$G = \frac{1}{N} \sum_i V_i.$$

Given the vector G, we represent $K(G)$ as a set of indices of the K largest entries. For each image I_i, we compute a weighted sum of feature maps of the last convolutional layer to get the CCAM according to the top K components.

$$M^i(h, w) = \sum_{k \in K(G)} G_k \sum_c W_c^k F_c(h, w). \tag{5.2}$$

To perform localization, we can generate a bounding box (see section 5.3.1) given the CCAM for the image I_i.

Using CCAM, we can decompose the neural activations of the common novel object into semantically interpretable components which are pre-trained with known object categories. In Fig. 5.3, the percentage of the contribution of each component and its corresponding known object class-specific CAM is shown.

5.3 Experiments

For a fair compassion with other approaches [86, 79], we evaluate the effect of our method using AlexNet [70], which is pre-trained by [153] using ILSVRC with 1000 image categories [108]. In the AlexNet, the penultimate fully-connected layer is replaced with a global average pooling layer. Our method can handle a set of images in linear time complexity $\mathcal{O}(N)$.

5.3.1 Generating Boxes

To produce a bounding box from CCAM, we use a similar threshold method as [153] to segment the heatmap. In particular, we segment the regions of which the value is above a fixed threshold. In contrast to [153], we only take a single box which covers the largest connected component in the segmentation map and includes the max value of the CCAM. In our experiment, we set the threshold to 25% of the max value of the CCAM. We take top $K = 200$ components for computing the CCAM.

5.3.2 Evaluation Metric

Following [27, 122], we use CorLoc as the evaluation metric, which is defined as the percentage of images in which a method correctly localizes the common objects. If there is one ground-truth box of the common object having more than 0.5 intersection-over-union with the predicted box, then we count this image as a correctly localized one.

	chipmunk	rhino	stoat	raccoon	rake	wheelchair	mean
Li *et al.* [86]	44.0	81.8	**67.3**	41.8	14.5	39.3	48.1
Le *et al.* [79]	44.9	**86.4**	56.7	**66.0**	10.3	32.4	49.5
Ours	**48.2**	77.9	55.7	57.3	**46.4**	**48.6**	**55.7**

Table 5.1: Object co-localization on subset of ImaegeNet.

5.3.3 Dataset

In order to evaluate our method for unseen object categories, we follow [86] and test our method on the six subsets of the ImageNet, which are not included in the ILSVRC. These unseen objects are *chipmunk*, *rhino*, *stoat*, *raccoon*, *rake*, and *wheelchair*.

5.3.4 Results

In Table. 5.1, we show the quantitative results of our method as well as the state-of-the-art approaches [86] and [79]. Clearly, our approach outperforms [86, 79] by a large margin. It is important to note that [86] use object proposal method and [79] use the over-segmentation method. Our method is proposal-free and superpixel-free. Visual examples of common object localization on the subset of the ImageNet can be found in Fig. 5.3. The ground-truth boxes are in red and the predicted boxes are in green.

5.4 Conclusion

In this chapter, we propose an approach to localize common objects from novel object categories in a set of images. We solve this problem by using CAMs as components instead of class-specific activation maps. As we show in the experiment section, our approach can localize novel object categories.

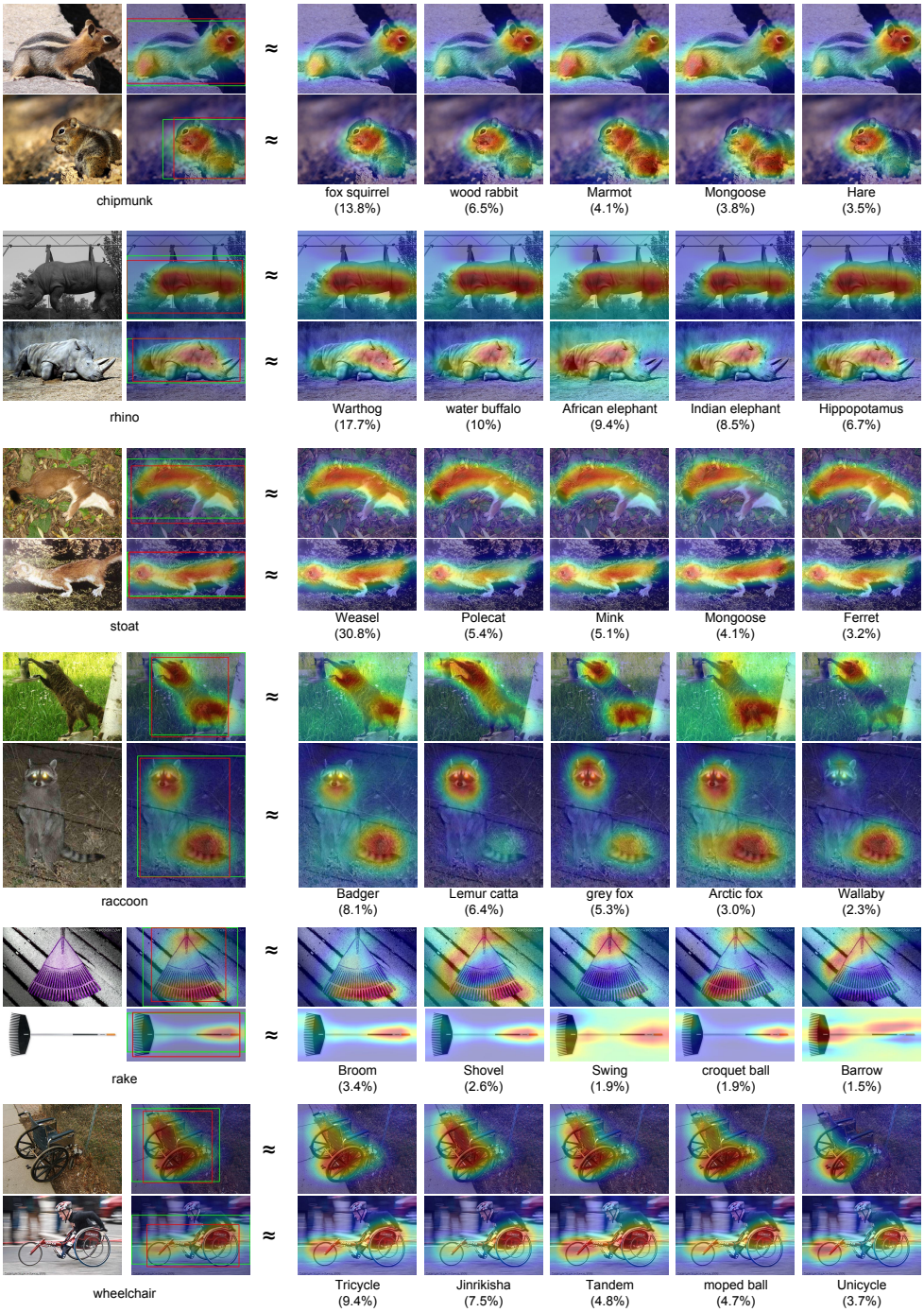

Figure 5.3: Viusal examples of common object localization on the subset of the ImageNet. Red boxes are ground-truth and green boxes are our predictions. Best viewed in colour.

Chapter 6

Conclusions

6.1 Overview

Semantic-aware image analysis is an important problem in computer vision with many applications. In this thesis, we explored four different aspects of image analysis:

- We explored the topic of semantic segmentation for the man-made scene using fully connected conditional random fields model.
- We proposed a semantic-aware image smoothing method by combining low-level, mid-level, and high-level vision features.
- We presented a new and efficient convolutional neural networks based method for solving the problem of object co-segmentation.
- We proposed an approach to localize common objects from novel object categories in a set of images using the common component activation map.

6.2 Outlook

While we did some progress in several aspects of image analysis, which included semantic segmentation, image smoothing, object co-segmentation, and object localization, there are still some limitations, further improvement, and open questions left to be addressed.

Man-Made Scenes Segmentation. In Chapter 2, since both the eTRIMS and LabelMeFacade image databases are relatively small, we train the Textonboost rather than CNNs as the unary classifier for each pixel. Recently, the success of CNNs in semantic segmentation is based on the availability of large annotated datasets. Generating synthetic man-made scenes data or combining the real and synthetic data [3] is a very promising direction for future work.

Semantic Smoothing. In Chapter 3, we used semantic segmentation as the high-level information structure prior for image smoothing. Extending our method by exploiting diverse levels of semantic information, such as instance segmentation, object part segmentation, and panoptic segmentation, is an interesting direction for future work. To get semantic information, we directly apply the off-the-shelf semantic segmentation methods. Another potential future work is jointly training a model for semantic segmentation and image smoothing.

Object Co-Segmentation. In Chapter 4, we proposed a simple and efficient CNN-based method for solving the problem of object co-segmentation. There are still two potential ways to further improve and generalize this task. First, in our work, we correlate the high-level features to detect the common objects. One promising future research is to employ multi-level features for hierarchical correction. Second, our object co-segmentation method tries to segment all of the common objects as foreground without any instance information. The co-segmentation can be extended to instance-level co-segmentation by resorting the region-based instance segmentation method, such as Mask R-CNN [44].

Object Co-Localization. In Chapter 5, we proposed a new CAM method to localize common objects in a set of images. The proposed method utilizes the attention of existing objects to approximate the attention of an unknown object, which is related to zero-shot learning. One very promising future research direction is zero-shot object localization.

Bibliography

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2274–2282, 2012.

[2] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. Fast high-dimensional filtering using the permutohedral lattice. *Comput. Graph. Forum*, 29(2):753–762, 2010.

[3] Hassan Abu Alhaija, Siva Karthik Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision*, 126(9):961–972, 2018.

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for scene segmentation. *TPAMI*, 2017.

[5] Min Bai, Wenjie Luo, Kaustav Kundu, and Raquel Urtasun. Exploiting semantic information and deep matching for optical flow. In *European Conference on Computer Vision (ECCV)*, pages 154–170. Springer, 2016.

[6] Jonathan T Barron and Ben Poole. The fast bilateral solver. In *European Conference on Computer Vision (ECCV)*, pages 617–632. Springer, 2016.

[7] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 2010.

[8] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3479–3487, 2015.

[9] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[10] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *ICCV*, 2001.

[11] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision*, pages 536–551. Springer, 2014.

[12] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and carsten Rother. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[13] Clemens-Alexander Brust, Sven Sickert, Marcel Simon, Erik Rodner, and Joachim Denzler. Efficient convolutional patch networks for scene understanding. In *CVPR Workshops*, 2015.

[14] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[15] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698, 1986.

[16] Joao Carreira and Cristian Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, 2010.

[17] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

[18] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. Enriching visual knowledge bases via object discovery and segmentation. In *CVPR*, 2014.

[19] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[20] Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi. Deep filter banks for texture recognition and segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3828–3836, 2015.

[21] Maxwell D Collins, Jia Xu, Leo Grady, and Vikas Singh. Random walks based multi-image segmentation: Quasiconvexity results and gpu-based solutions. In *CVPR*, 2012.

[22] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.

[23] Antonio Criminisi, Toby Sharp, Carsten Rother, and Patrick P'erez. Geodesic image and video editing. *ACM Trans. Graph.*, 29(5):134:1–134:15, November 2010.

[24] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[25] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 886–893, 2005.

[26] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[27] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Localizing objects while learning their appearance. In *European conference on computer vision*, pages 452–466. Springer, 2010.

[28] Piotr Dollár, Zhuowen Tu, Pietro Perona, and Serge J. Belongie. Integral channel features. In *British Machine Vision Conference, BMVC 2009, London, UK, September 7-10, 2009. Proceedings*, pages 1–11, 2009.

[29] Piotr Dollár and C Lawrence Zitnick. Structured forests for fast edge detection. In *International Conference on Computer Vision (ICCV)*, pages 1841–1848, 2013.

[30] Xingping Dong, Jianbing Shen, Ling Shao, and Ming-Hsuan Yang. Interactive cosegmentation using global and local energy optimization. *IEEE Transactions on Image Processing*, 2015.

[31] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015.

[32] Elmar Eisemann and Frédo Durand. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph.*, 23(3):673–678, August 2004.

[33] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[34] Alon Faktor and Michal Irani. Co-segmentation by composition. In *ICCV*, 2013.

[35] Zeev Farbman, Raanan Fattal, and Dani Lischinski. Diffusion maps for edge-aware image editing. *ACM Trans. Graph.*, 29(6):145:1–145:10, December 2010.

[36] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Trans. Graph.*, 27(3):67:1–67:10, August 2008.

[37] Björn Fröhlich, Erik Rodner, and Joachim Denzler. A fast approach for pixelwise labeling of facade images. In *20th International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey, 23-26 August 2010*, pages 3029–3032, 2010.

[38] Huazhu Fu, Dong Xu, Stephen Lin, and Jiang Liu. Object-based rgbd image co-segmentation with mutex constraint. In *CVPR*, 2015.

[39] Eduardo S. L. Gastal and Manuel M. Oliveira. Domain transform for edge-aware image and video processing. *ACM Trans. Graph.*, 30(4):69:1–69:12, July 2011.

[40] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[41] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[42] Christian Hane, Christopher Zach, Andrea Cohen, Roland Angst, and Marc Pollefeys. Joint 3d scene reconstruction and class segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 97–104, 2013.

[43] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011.

[44] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[45] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In *European Conference on Computer Vision (ECCV)*, pages 1–14. Springer, 2010.

[46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[48] Dorit S Hochbaum and Vikas Singh. An efficient algorithm for co-segmentation. In *ICCV*, 2009.

[49] Wenqi Huang, Xiaojin Gong, and Michael Ying Yang. Joint object segmentation and depth upsampling. *IEEE Signal Processing Letters*, 22(2):192–196, 2015.

[50] Omid Hosseini Jafari, Oliver Groth, Alexander Kirillov, Michael Ying Yang, and Carsten Rother. Analyzing modular cnn architectures for joint depth prediction and semantic segmentation. In *International Conference on Robotics and Automation (ICRA)*, 2017.

[51] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Pixel objectness. *arXiv:1701.05349*, 2017.

[52] Varun Jampani, Raghudeep Gadde, and Peter V. Gehler. Efficient facade segmentation using auto-context. In *WACV*, pages 1038–1045, 2015.

[53] Koteswar Rao Jerripothula, Jianfei Cai, Fanman Meng, and Junsong Yuan. Automatic image co-segmentation using geometric mean saliency. In *ICIP*, 2014.

[54] Koteswar Rao Jerripothula, Jianfei Cai, and Junsong Yuan. Image co-segmentation via saliency co-fusion. *IEEE Transactions on Multimedia*, 2016.

[55] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*, 2014.

[56] Di Jin, Ziyang Liu, Weihao Li, Dongxiao He, and Weixiong Zhang. Graph convolutional networks meet markov random fields: Semi-supervised community detection in attribute networks. In *AAAI*, 2019.

[57] Di Jin, Xinxin You, Weihao Li, Dongxiao He, Peng Cui, Françoise Fogelman-Soulié, and Tanmoy Chakraborty. Incorporating network embedding into markov random field for better community detection. In *AAAI*, 2019.

[58] Armand Joulin, Francis Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010.

[59] Armand Joulin, Kevin Tang, and Li Fei-Fei. Efficient image and video co-localization with frank-wolfe algorithm. In *European Conference on Computer Vision*, pages 253–268. Springer, 2014.

[60] Jörg H. Kappes, Björn Andres, Fred A. Hamprecht, Christoph Schnörr, Sebastian Nowozin, Dhruv Batra, Sungwoong Kim, Bernhard X. Kausler, Thorben Kröger, Jan Lellmann, Nikos Komodakis, Bogdan Savchynskyy, and Carsten Rother. A comparative study of modern inference techniques for structured discrete energy minimization problems. *International Journal of Computer Vision*, 115(2):155–184, 2015.

[61] Levent Karacan, Erkut Erdem, and Aykut Erdem. Structure-preserving image smoothing via region covariances. *ACM Trans. Graph.*, 32(6):176:1–176:11, November 2013.

[62] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[63] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. *arXiv preprint arXiv:1901.02446*, 2019.

[64] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[65] Pushmeet Kohli, Lubor Ladicky, and Philip H. S. Torr. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82(3):302–324, 2009.

[66] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[67] F. Korč and W. Förstner. eTRIMS Image Database for interpreting images of man-made scenes. Technical Report TR-IGG-P-2009-01, April 2009.

[68] Adarsh Kowdle, Dhruv Batra, Wen-Chao Chen, and Tsuhan Chen. imodel: Interactive co-segmentation for object of interest 3d modeling. In *ECCV workshop*, 2010.

[69] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, pages 109–117, 2011.

[70] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[71] Alexander Krull, Eric Brachmann, Frank Michel, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Learning analysis-by-synthesis for 6d pose estimation in rgb-d images. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[72] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[73] Abhijit Kundu, Yin Li, Frank Dellaert, Fuxin Li, and James M Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In *European Conference on Computer Vision (ECCV)*, pages 703–718. Springer, 2014.

[74] Jan Eric Kyprianidis and Jürgen Döllner. Image abstraction by structure adaptive filtering. In *TPCG*, pages 51–58.

[75] Lubor Ladicky, Christopher Russell, Pushmeet Kohli, and Philip H. S. Torr. Associative hierarchical crfs for object class image segmentation. In *ICCV*, pages 739–746, 2009.

[76] Lubor Ladickỳ, Paul Sturgess, Chris Russell, Sunando Sengupta, Yalin Bastanlar, William Clocksin, and Philip HS Torr. Joint optimization for object class segmentation and dense stereo reconstruction. *International Journal of Computer Vision*, 100(2):122–133, 2012.

[77] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001.

[78] Måns Larsson, Fredrik Kahl, Shuai Zheng, Anurag Arnab, Philip H. S. Torr, and Richard I. Hartley. Learning arbitrary potentials in crfs with gradient descent. *CoRR*, abs/1701.06805, 2017.

[79] Hieu Le, Chen-Ping Yu, Gregory Zelinsky, and Dimitris Samaras. Co-localization with category-consistent features and geodesic distance propagation. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[80] Chulwoo Lee, Won-Dong Jang, Jae-Young Sim, and Chang-Su Kim. Multiple random walkers and their application to image cosegmentation. In *CVPR*, 2015.

[81] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In *CVPR*, 2016.

[82] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep object co-segmentation. In *ACCV*, 2018.

[83] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Localizing common objects using common component activation map. In *CVPR Workshop on Explainable AI*, 2019.

[84] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Semantic-aware image smoothing. In *Proceedings of the conference on Vision, Modeling and Visualization*, pages 153–160. Eurographics Association, 2017.

[85] Weihao Li and Michael Ying Yang. Efficient semantic segmentation of man-made scenes using fully-connected conditional random field. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41:633, 2016.

[86] Yao Li, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. Image co-localization by mimicking a good detector's confidence score distribution. In *European Conference on Computer Vision*, pages 19–34. Springer, 2016.

[87] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017.

[88] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[89] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[90] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[91] Andelo Martinovic and Luc J. Van Gool. Bayesian grammar learning for inverse procedural modeling. In *CVPR*, pages 201–208, 2013.

[92] Andelo Martinovic, Markus Mathias, Julien Weissenberg, and Luc J. Van Gool. A three-layered approach to facade parsing. In *ECCV*, pages 416–429, 2012.

[93] Frank Michel, Alexander Kirillov, Eric Brachmann, Alexander Krull, Stefan Gumhold, Bogdan Savchynskyy, and Carsten Rother. Global hypothesis generation for 6d object pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[94] Lopamudra Mukherjee, Vikas Singh, and Charles R Dyer. Half-integrality based algorithms for cosegmentation of images. In *CVPR*, 2009.

[95] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015.

[96] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[97] Sylvain Paris, Samuel W. Hasinoff, and Jan Kautz. Local laplacian filters: Edge-aware image processing with a laplacian pyramid. *ACM Trans. Graph.*, 30(4):68:1–68:12, July 2011.

[98] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[99] Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph.*, 23(3):664–672, August 2004.

[100] Rong Quan, Junwei Han, Dingwen Zhang, and Feiping Nie. Object co-segmentation via graph optimized-flexible manifold ranking. In *CVPR*, 2016.

[101] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[102] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[103] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[104] Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *CVPR*, 2006.

[105] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, 2013.

[106] Jose C Rubio, Joan Serrat, Antonio López, and Nikos Paragios. Unsupervised co-segmentation through region matching. In *CVPR*, 2012.

[107] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, November 1992.

[108] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[109] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

[110] Nick Schneider, Lukas Schneider, Peter Pinggera, Uwe Franke, Marc Pollefeys, and Christoph Stiller. Semantically guided depth upsampling. In *German Conference on Pattern Recognition (GCPR)*, pages 37–48. Springer, 2016.

[111] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[112] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv:1312.6229*, 2013.

[113] Laura Sevilla-Lara, Deqing Sun, Varun Jampani, and Michael J Black. Optical flow with semantic segmentation and localized layers. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3889–3898, 2016.

[114] Tong Shen, Guosheng Lin, Lingqiao Liu, Chunhua Shen, and Ian Reid. Weakly supervised semantic segmentation based on co-segmentation. In *BMVC*, 2017.

[115] Xiaoyong Shen, Chao Zhou, Li Xu, and Jiaya Jia. Mutual-structure for joint filtering. In *International Conference on Computer Vision (ICCV)*, pages 3406–3414, 2015.

[116] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[117] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

[118] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[119] Richard Socher, Cliff Chiung-Yu Lin, Andrew Y. Ng, and Christopher D. Manning. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 129–136, 2011.

[120] Kartic Subr, Cyril Soler, and Frédo Durand. Edge-preserving multiscale image decomposition based on local extrema. *ACM Trans. Graph.*, 28(5):147:1–147:9, December 2009.

[121] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[122] Kevin Tang, Armand Joulin, Li-Jia Li, and Li Fei-Fei. Co-localization in real-world images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[123] Tatsunori Taniai, Sudipta N. Sinha, and Yoichi Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *CVPR*, 2016.

[124] Olivier Teboul, Iasonas Kokkinos, Loïc Simon, Panagiotis Koutsourakis, and Nikos Paragios. Shape grammar parsing via reinforcement learning. In *CVPR*, pages 2273–2280, 2011.

[125] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *International Conference on Computer Vision (ICCV)*, pages 839–846. IEEE, 1998.

[126] Zhuowen Tu. Auto-context and its application to high-level vision tasks. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*, 2008.

[127] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 2013.

[128] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Cosegmentation revisited: Models and optimization. In *ECCV*, 2010.

[129] Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object cosegmentation. In *CVPR*, 2011.

[130] Vibhav Vineet, Carsten Rother, and Philip Torr. Higher order priors for joint intrinsic image, objects, and attributes estimation. In *Neural Information Processing Systems (NIPS)*, pages 557–565, 2013.

[131] Vibhav Vineet, Jonathan Warrell, and Philip H. S. Torr. Filter-based mean-field inference for random fields with higher-order terms and product label-spaces. *International Journal of Computer Vision*, 110(3):290–307, 2014.

[132] Fan Wang, Qixing Huang, and Leonidas J. Guibas. Image co-segmentation via consistent functional maps. In *ICCV*, 2013.

[133] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, 2016.

[134] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L. Yuille. Joint object and part segmentation using deep learned potentials. In *International Conference on Computer Vision (ICCV)*, December 2015.

[135] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Towards unified depth and semantic prediction from a single image. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2800–2809, 2015.

[136] Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia. Image smoothing via l0 gradient minimization. *ACM Trans. Graph.*, 30(6):174:1–174:12, December 2011.

[137] Li Xu, Jimmy Ren, Qiong Yan, Renjie Liao, and Jiaya Jia. Deep edge-aware filters. In *International Conference on Machine Learning (ICML)*, pages 1669–1678, 2015.

[138] Li Xu, Qiong Yan, Yang Xia, and Jiaya Jia. Structure extraction from texture via relative total variation. *ACM Trans. Graph.*, 31(6):139:1–139:10, November 2012.

[139] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas Huang. Deep grabcut for object selection. In *BMVC*, 2017.

[140] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S. Huang. Deep interactive object selection. In *CVPR*, 2016.

[141] Michael Ying Yang and Wolfgang Förstner. A hierarchical conditional random field model for labeling and classifying images of man-made scenes. In *ICCV Workshops*, pages 196–203, 2011.

[142] Michael Ying Yang, Wolfgang Förstner, and Martin Drauschke. Hierarchical conditional random field for multi-class image classification. In *International Conference on Computer Vision Theory and Applications (VISSAPP)*, pages 464–469, 2010.

[143] Qingxiong Yang. Semantic filtering. In *Computer Vision and Pattern Recognition (CVPR)*, pages 4517–4526, 2016.

[144] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[145] Zehuan Yuan, Tong Lu, and Yirui Wu. Deep-dense conditional random fields for object co-segmentation. In *IJCAI*, 2017.

[146] Feihu Zhang, Longquan Dai, Shiming Xiang, and Xiaopeng Zhang. Segment graph based image filtering: Fast structure-preserving smoothing. In *International Conference on Computer Vision (ICCV)*, pages 361–369, 2015.

[147] Qi Zhang, Xiaoyong Shen, Li Xu, and Jiaya Jia. Rolling guidance filter. In *European Conference on Computer Vision (ECCV)*, pages 815–830. Springer, 2014.

[148] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S. Huang. Adversarial complementary learning for weakly supervised object localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[149] Xiaolin Zhang, Yunchao Wei, Guoliang Kang, Yi Yang, and Thomas Huang. Self-produced guidance for weakly-supervised object localization. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[150] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.

[151] Shuai Zheng, Ming-Ming Cheng, Jonathan Warrell, Paul Sturgess, Vibhav Vineet, Carsten Rother, and Philip HS Torr. Dense semantic image segmentation with objects and attributes. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3214–3221, 2014.

[152] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015.

[153] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.