Holistic Image Understanding with Deep Learning and Dense Random Fields

Shuai Zheng

St Catherine's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2016

This thesis is dedicated to
my family
for their tremendous support and love.

Acknowledgements

I would like to thank my supervisor Professor Philip Torr for showing me the concept of science and the art of research. Without his guidance, I would hardly know where to start to do good, solid research. I would like to thank individually everyone that has helped me through the course of my DPhil. This thesis stems from exciting collaborations with Professor Carsten Rother, Ming-Ming Cheng, Professor Niloy Mitra, Jamie Shotton, Bernardino Romera-Paredes, Sadeep Jayasumana, Jonathan Warrell, Paul Sturgess, Wen-Yan Lin, Professor Nigel Crook, Vibhav Vineet, Chang Huang, Zhizhong Su, and Dalong Du. I would also like to thank my other collaborators during the last stage of my DPhil: Professor Andrea Vedaldi, Anurag Arnab, Melinos Averkiou, and James Thewlis. We had lots of fruitful research results in the end. I profoundly thank my transfer report examiners Professor Andrew Zisserman, and Professor Michael Osborne, who helped me to improve my DPhil research plan overtime. I also would like to thank my examiners Dr. Patrick Perez and Professor David Murray for their suggestions and discussion on this thesis.

I would like to thank all other lab-mates: Carl, Olaf, Duncan, Stuart, Jack, Eric, Michael, Siddharth, Julien, Morten, Ondrej, Luca, Saumya, Rodrigo, Jack Hunt, Daniela, Rudy, Oscar, Ziming, Sunando, Glenn, Sam, Paul, Natasha, Diane, Alban, Pankaj, Lubor, Professor Pawan Kumar, Omkar, Weilin, Elliot, Karel, Aravindh, David, Yujie, Tomas, Relja, Yusuf, Yuning, Ken, Mircea, Max, Meelis, Minh, Lyndsey, Karen, Siddharth, Ankush, and all ex-members. They made my days in Oxford very special and enjoyable.

I would also thank Professor Carsten Rother for hosting me twice in his lab in TU Dresden. From these visits, I not only developed deeper knowledge in discrete optimization but also made lots of friends: Michael Hornáček, Uwe Schmidt, Alexander Krull, Eric Brachmann, Frank Michel, Dmitrij Schlesinger, Alexander Zouhar, Alexander Kirillov, Michael Yang, Michael Kalygin, Michael Figurnov, and many others.

During my DPhil, I had a research internship at Baidu Institute of deep learning. Thanks to my mentor Chang Huang, and my collaborators Dalong Du and Zhizhong Su, it was an extraordinary research experience.

I would like to thank Rudy Bunel, Anurag Arnab, Bernardino Romera-Paredes, Michael Sapienza, and Stuart Golodetz for their great help proof-reading this thesis.

I am immensely grateful to EPSRC and ERC for their generous financial support, without which my studies would not have been possible. I would also like to thank the IEEE CVPR, Oxford University Engineering Science Department and St Catherine's College for supporting my attendance at conferences. Last, but not least, I express my thanks to my family for their never-faltering support, encouragement, and love.

Abstract

One aim of holistic image understanding is not only to recognise the things and stuff in images but also to localise where they are exactly. Semantic image segmentation is set up to achieve this goal. The purpose of this task is to recognise and delineate the visual objects. The solution to this task provides detailed information to understand images and is central to applications such as content-based image search, autonomous vehicles, image-editing, and smart glasses for partially-sighted people. This task is challenging to address not only because the visual objects from the same category could have a variety of appearances but also because of the need to account for contextual information across images such as edges and appearance consistency. The objective of this thesis is to bridge the gap between the pixel-based image representation in computer devices and the meaning extracted by humans.

Our primary contributions are fourfold. Firstly, we propose a factorial fully-connected conditional random field that addresses the problem of jointly estimating the segmentation for both object class and visual attributes. Secondly, we embed the proposed factorial fully-connected conditional random fields framework in an interactive image segmentation system. This system allows users to refine the semantic image segmentation with verbal instructions. Thirdly, we formulate filter-based mean-field approximate inference for fully-connected conditional random fields with Gaussian pairwise potentials as a recurrent neural network. This formulation allows us to integrate fully convolutional neural networks and conditional random fields in an end-to-end trainable system. Fourthly, we show the relationship between fully-connected conditional random fields with Gaussian pairwise potentials and iterative Graph-cut: We found that fully-connected conditional random fields with Gaussian Pairwise potential implicitly model the unnormalised global colour models for foreground and background.

Contents

List of Figures

Chapter 1

Introduction

1.1 Objective

The objective of this thesis is to propose new techniques to recognize objects in images and delineate their 2D outlines. Humans describe images in terms of language components such as nouns (*e.g.* bed, cupboard, desk) and adjectives (*e.g.* textured, wooden), while pixels form a natural representation for computer devices. Bridging this gap between how humans would like to access images versus their typical computerised representation is the goal of this thesis. In particular, we address the problem of semantic image segmentation, and its extensions such as semantic image segmentation with objects and visual attributes, and interactive image segmentation. These tasks are illustrated in figure 1, and described as follows:

Semantic Image Segmentation aims to partition an image into coherent regions and determine semantically-meaningful labels for each region.

Semantic Image Segmentation with objects and visual attributes address the problem of jointly assigning both object class labels (*e.g.* bed) and visual attributes (*e.g.* wooden) to each pixel in the images.

Interactive image segmentation aims to delineate particular objects of interest from images with a small amount of human aid, such as verbal instruction.

1.2 Motivation

Visual perception has played essential role for humans to survive and evolve. Most healthy humans take for granted the ability to see the world and understand complex pictures. In contrast, computer devices only see pictures as a set of pixels. Empowering machines to see the world as healthy humans do would not only create artificial intelligent for robots but also could be used to improve aids for those who have a visual impairment. In fact,

for those people who are suffering from imperfect vision, helping them to be able to live as normal remains a technique challenging problem. Helping robots and the visually impaired to see the world is the main inspiration for this thesis. The semantic image segmentation techniques developed in this thesis would be useful to the assistant applications for both robotics and the visually impaired although we do not attempt to develop visual assistants. We describe the potential applications of our techniques as follows.

Assistant for the visually impaired: From ancient times till now, it has always been difficult to live with poor sight or without sight at all. Being able to see helps us to perform the essential actions in life, for example, navigating from home to work, avoiding harm and recognizing food. Losing eyesight decreases the ability to live independently. There are millions of people in the world nowadays who are suffering from visual impairment. According to the RNIB[1], there are almost two million people in the UK who are suffering from sight loss. They would also be useful for further developing the smart glasses for the partially visual impaired [150]. Being able to interpret images with objects and visual attributes would also help the blind by converting the visual information to audio.

Robotics: Computer vision techniques play a significant role in developing robust robotics. In particular, an autonomous driving car would use our semantic image segmentation techniques to see a potential hazard and also find out where is the road. Semantic image segmentation technologies developed in this thesis would help robots to recognize everyday objects and delineate the 2D outline of them.

Heathcare & Medical: Discovering and delineating tumors is tedious and requires a lot of skill in medical research. Semantic image segmentation techniques developed here would be useful when integrated into an automatic system to speed up this process and cut down on human errors.

Intelligent visual surveillance: The ability to automatically recognize suspicious individuals or terrorists would help in the criminal investigations and could save lives. Together with other technologies such as face recognition and detection, semantic image segmentation provides more detailed and precise scene understanding. This would also essentially help to reduce false alarms. Intelligent visual surveillance systems would greatly benefit from the semantic image segmentation techniques described in this thesis.

National security: Aerial images provide essential information for the interest of national security. The advance of semantic image segmentation would help to develop better way of automatic extracting the road, building and objects of interests from aerial images. This would make the aerial image analysis software more robust and efficient.

[1] http://www.rnib.org.uk/knowledge-and-research-hub/key-information-and-statistics

Image editing: As we describe later, the semantic image segmentation techniques developed in this thesis would help to segment the objects of interest from pictures, and users can provide verbal instructions to further refine semantic image segmentation results. This technology would be useful for devices like mobile phones, tablets, and television, where precise mouse controls are not available.

E-commerce: E-commercial platforms like eBay, Alibaba, and Amazon connect millions of buyers and sellers. The smart mobile phone provides a powerful tool to take pictures and videos. The techniques developed in this thesis would potentially improve the user experience of selling and buying stuff on these e-commerce applications. For example, on those second-hand trading platforms, with our techniques, the system would automatically recognize and segment the products from the images uploaded by sellers, and then generate pre-filled forms for the seller. The proposed technologies would potentially help to save a lot of time for a new vendor.

Although all applications mentioned above require careful development and further integration, accurate and efficient semantic image segmentation techniques can make a big difference in these areas.

1.3 Challenges

Semantic image segmentation faces many challenges when deployed into real-world applications.

Appearance variations: Semantic image segmentation consists of category-based object recognition and image reorganization. Object recognition is a challenging problem itself. The objects from the same category might present notable appearance difference from various viewpoints, poses, or lighting conditions. They might also be partially visible due to occlusion or environment factors. These challenges require a robust feature representation. The state-of-the-art object recognition systems significantly benefit from the use of large-scale Convolutional Neural Networks, and this would apply for semantic image segmentation as well.

Lack of context and global information: Context information is important for various recognition related computer vision tasks such as object detection. It is important to develop solutions to explore the contextual information to achieve the state-of-the-art semantic image segmentation performance. In this thesis, we propose an approach that integrates deep convolutional neural networks and conditional random fields. The latter helps the system to capture the longer connectivity and context in image segmentation.

Lack of large-scale and high-quality annotations: Current state-of-the-art semantic image segmentation systems are using supervised learning approaches. They are often pre-trained on the ImageNet [58] Classification data set and then fine-tuned on high-quality segmentation annotated data sets, such as PASCAL VOC Challenge [68] data set, and the CityScape [50] data set. For many new problems, it is very challenging to efficiently label the high-quality image segmentation ground truth.

1.4 Approach

To address the problem of semantic image segmentation, we consider the approaches that combine modern feature representation learning approaches and Conditional Random Fields (CRFs).

The feature representation learning methods such as Deep Convolutional neural networks (CNNs) learn the feature representation and pixel-wise classifiers in a data-driven way. The most significant difference between the TextonBoost [192] and CNNs is that CNNs learn both features representation and classifiers in an end-to-end fashion, while the traditional methods like TextonBoost use the hand-craft engineered features (*e.g.* SIFT, LBP, Texton, etc.) and learn the pixel-wise classifiers separately. In this thesis, we explore both options. We find that integrating the CNNs and a particular type of fully-connected CRFs result in significant performance improvements.

1.5 Contributions

The main contributions of this thesis are fourfold:

- We propose a factorial fully-connected conditional random fields framework that could address the problem of jointly estimating the segmentation for both object class and visual attributes.
- We show that our proposed factorial fully-connected CRFs can be tailored in an interactive image segmentation system with verbal instructions, resulting in a significant improvement over automatic semantic image segmentation.
- We investigate the connections between deep Convolutional Neural Networks and conditional random fields. We found that the mean-field approximate inference for fully-connected CRFs can be reformulated as a series of CNN operations, and we could further form an end-to-end trainable semantic image segmentation composed of both CNNs and CRFs.

- We found that a fully-connected conditional random fields with Gaussian Pairwise potentials implicitly models unnormalised global colour models for foreground and background. This provides insightful analysis to bridge the filter-based variational mean-field approximate inference and the iterative Graph-cut inference functionality.

1.6 Publications

Chapter 2 through Chapter 5 are based on the following published works, which has been editing for the purpose of completeness and consistency.

[1] Ming-Ming Cheng, Victor Adrian Prisacariu, Shuai Zheng, Philip H.S. Torr, and Carsten Rother. Densecut: densely connected CRFs for realtime GrabCut. *Computer Graphics Forum*, 34(7), 2015.

[2] Shuai Zheng, Ming-Ming Cheng, Wen-Yan Lin, Jonathan Warrell, Vibhav Vineet, Paul Sturgess, Nigel Crook, Nioly Mitra, and Philip H. S. Torr. ImageSpirit: Verbal Guided Image Parsing. *ACM TOG*, 2014.

[3] Shuai Zheng, Ming-Ming Cheng, Jonathan Warrell, Paul Sturgess, Vibhav Vineet, Carsten Rother, and Philip .H. S. Torr. Dense semantic image segmentation with objects and attributes. In *IEEE CVPR*, 2014.

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

During my DPhil, I am grateful to work with other colleagues, have published and coauthor in the following papers.

[1] Anurag Arnab, Sadeep Jayasumana, Shuai Zheng, and Philip Torr. Higher order conditional random fields in deep neural networks. In *ECCV*, 2016.

[2] Ming-Ming Cheng, Jonathan Warrell, Wen-Yan Lin, Shuai Zheng, Vibhav Vineet, and Nigel Crook. Efficient salient region detection with soft image abstraction. In *IEEE ICCV*, 2013.

[3] Alexander Kirillov, Dmitrij Schlesinger, Shuai Zheng, Bogdan Savchynskyy, Philip Torr, and Carsten Rother. Efficient likelihoood learning of a generic cnn-crf model for semantic segmentation. In *ACCV*, 2016.

[4] Wen-Yan Lin, Ming-Ming Cheng, Shuai Zheng, J. Lu, and N. Crook. Robust non-parametric data fitting for correspondence modeling. In *IEEE ICCV*, 2013.

[5] James Thewlis, Shuai Zheng, Philip H. S. Torr, and Andrea Vedaldi. Fully trainable deep matching. In *BMVC*, 2016.

[6] Shuai Zheng, Victor Adrian Prisacariu, Melinos Averkiou, Ming-Ming Cheng, Niloy Mitra, Jamie Shotton, Philip H. S. Torr, and Carsten Rother. Object proposal estimation in depth images using compact 3D shape manifolds. In *German Conference on Pattern Recognition*, 2015.

[7] Shuai Zheng, Paul Sturgess, and Philip H. S. Torr. Approximate structured output learning for constrained local models with application to real-time facial feature detection and tracking on low-power devices. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2013.

1.7 Outline

This thesis consists of five chapters. The first is this introduction. In Chapter 2, we propose the factorial fully-connected CRFs for semantic image segmentation with objects and visual attributes. In Chapter 3, we show that we can integrate our factorial fully-connected CRFs into an interactive image segmentation system with verbally guided instruction. In Chapter 4, we propose an end-to-end trainable semantic image segmentation system that integrates deep CNNs and a fully-connected CRFs with Gaussian pairwise potentials. In Chapter 5, we describe DenseCut, an efficient substitute foreground segmentation method based on a fully-connected CRFs with Gaussian pairwise potentials.

In the appendix, we have included several tutorials about the basic knowledge related to the topic in this thesis, and we have also summarised the state-of-the-art for image segmentation. We present a tutorial about filter-based mean-field approximate inference in appendix A. We also briefly summarize the Convolutional and Deconvolution operations in appendix B. We discuss the Recurrent Neural Network in appendix C. We review the works related to semantic image segmentation in appendix D.

Chapter 2

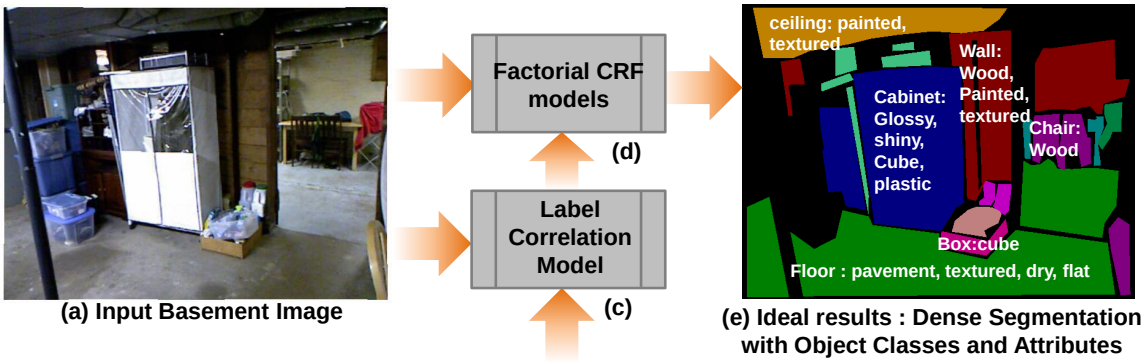
Dense Semantic Image Segmentation with Objects and Attributes

> I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have "327"? No. I have sky, house, and trees.
>
> *Max Wertheimer*

The concepts of objects and attributes are both important for describing images precisely since verbal descriptions often contain both adjectives and nouns (e.g. 'I see a shiny red chair'). In this chapter, we formulate the problem of joint visual attribute and object class image segmentation as a dense multi-labelling problem, where each pixel in an image can be associated with both an object class and a set of visual attributes labels. To learn the label correlations, we adopt a boosting-based piecewise training approach to determine the relationships between the visual appearance and co-occurrence cues. We use a filter-based mean-field approximation method for efficient joint inference. Further, we develop a hierarchical model to incorporate region-level object and attribute information. Experiments on the aPASCAL, CORE, and attribute-augmented NYU indoor scenes datasets show that the proposed approach can achieve state-of-the-art results.

2.1 Introduction

Using objects and attributes jointly provides a much more precise way to describe the content of a scene than using only one alone. *e.g.* , the image description *a shiny red chair* is more precise than the description *chair* on its own. Motivated by this fact, we introduce the problem of joint attribute-object image segmentation, where each image pixel is labelled



(a) Input Basement Image

(e) Ideal results : Dense Segmentation with Object Classes and Attributes

	Objects	Attributes
Region-Level	Things(e.g. Chair, TV monitor, Box, etc.)	Shape (e.g. Cube) , Material, Surface …
Pixel-Level	Things and stuff (e.g. Chair, TV monitor, Box, Wall, ceiling, floor, etc.)	Material (e.g. plastic, wood), Surface (e.g. glossy), …

(b) Semantic Space

Figure 2.1: **Illustration of the proposed approach.** (a) shows the input image, a scene image from NYU dataset. (b) represents the semantic label space including pixel-level objects and attributes, region-level objects and region attributes. (e) shows conceptual ideal results for dense semantic segmentation with objects and attributes. Best view in colour.

with (i) an object label, such as car or road, (ii) visual attribute labels such as materials (wood, glass), and (iii) surface properties (shiny, glossy). We also make the distinction between things and stuff; where objects with a well defined shape and centroid are called things, and amorphous objects are refered to as stuff [91, 103, 117]. This problem is well suited for being solved in a joint hierarchical model, as the attributes can help with the object predictions and vice versa in both region and pixel levels.

In semantic image segmentation for object classes, existing approaches, *e.g.* [116, 192], treat the problem as a multi-class classification problem, where the goal is to associate each pixel with one of the object class labels. Recent works have also shown the advantages of using *visual attributes* [72, 76, 121, 186] and *relative visual attributes* [158] in object recognition, object localization [121, 73, 226], and scene classification [162, 231]. However, few of these works have been proposed to address the problem of dense image segmentation for things and stuff using attributes, and it is not yet clear whether visual attributes improve the performance of object segmentation.

In this chapter, we model scene images using a fully-connected multi-label conditional random field (CRF) with joint learning and inference. In our framework each image pixel is associated with both a set of attributes and a single object-class label, as illustrated in Fig. 2.1. In order to efficiently tackle the multi-labelling problem, we break it down into

manageable multi-class and binary subproblems using a factorial CRF framework [104, 119, 200]. The structure of the factorial CRF we propose includes links between object and attribute factors that explicitly allow us to model correlations between these output variables. In order to handle the use of attributes at different levels, we also propose a hierarchical model in which both objects and attributes are labelled at two levels, pixels and regions. Using the regions provided by the efficient object detector [4, 47, 74, 216] and the segmentation methods [21, 44, 46, 178], we can predict attributes such as shape, which apply to object instances as a whole. This allows us to deal with attributes both for objects of fixed spatial extent, *i. e.* things that can be detected with deformable part based detector (*e.g.* chair, etc) as well as amorphous objects (stuff), *i. e.* ones that are more ambiguous (*e.g.* floor, etc). Previous works [71, 72] have only focused on one of these forms and have not attempted to solve both types. To learn the correlations between factors we employ a boosting framework [183, 187] that exploits both the visual similarity and co-occurrence relations between object and attributes labels. This provides an effective piecewise learning strategy to train the model. To perform joint inference we use a mean field based algorithm [110, 223]. This allows us to use a fully-connected graph topology for both object and attribute factor CRFs, whilst maintaining efficiency through filtering.

Our work is different from previous works [83, 206] in several ways. Both these approaches deal only with a very limited set of spatial attributes. While Tighe *et al.* [206] consider a region MRF with only adjacent pairwise connections, we propose a hierarchical model with both pixel and region levels, which is fully-connected at the pixel level. We also use mean-field inference rather than graph-cuts to handle the dense topology. Gould *et al.* [83] only consider pixel labelling for object classes and spatial attributes. In contrast, our approach can deal with a much more general problem. Furthermore, we also differ substantially from [239]. They have also considered the task of estimating objects and attributes in images. However the focus of that work is to analyse the use of verbal interactions, performed by the user, in order to verbally guide image editing. They have not explored a hierarchical formulation, as done in this work, which is important to achieve a higher level of accuracy. Also, they have not considered learning the attribute-object relationship using a boosting-based piecewise training.

Our contributions in this chapter are as follows:

- We present an efficient hierarchical fully-connected multi-label CRF based framework, which involves assigning pixels with object class and attributes labels.
- We explore a piecewise boosting-based training strategy to learn the label correlations based on visual appearance similarity and label co-occurrence statistics.

- We augment the NYU dataset [193] with attribute labels (*attribute* NYU *dataset*, aNYU) to provide a benchmark to encourage alternative approaches.

2.2 Factorial Multi-Label CRF Model

We address the problem of joint semantic image segmentation for objects and attributes using a multi-label CRF, which we factor into multi-class and binary CRFs. Table 2.1 shows the list of notations through this chapter.

2.2.1 Multi-class CRF for Objects

We first review a general multi-class CRF model, which we will use as a factor in the joint model for the object classes, and which we generalize below to form the multi-label CRF for attribute labels. We define the CRF over a set of random variables, $\mathcal{X} = \{X_1, X_2, ..., X_N\}$, where each variable will take values from a set of *object labels*, $x_i \in \mathcal{O}$, where $\mathcal{O} = \{l_1, l_2, ..., l_K\}$. We denote by $\mathbf{x}$ a joint configuration of these random variables, and write $\mathbf{I}$ for the observed image data. The random field is defined over a graph $G(\mathcal{V}, \mathcal{E})$ with the i-th vertex being associated with a corresponding X_i and $(i, j) \in \mathcal{E}$ representing the i-th vertex and the jth vertex are connected by an edge. A pairwise multi-class CRF model can be defined in terms of an energy function:

$$E^{\mathcal{O}}(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i^{\mathcal{O}}(x_i) + \sum_{\{i,j\} \in \mathcal{E}} \psi_{ij}^{\mathcal{O}}(x_i, x_j), \tag{2.1}$$

where $\psi_i^{\mathcal{O}}$ and $\psi_{ij}^{\mathcal{O}}$ are potential functions discussed below. The probability of a configuration $\mathbf{x}$ under the CRF distribution is found by normalising the exponential of its negative energy, $P(\mathbf{x}|\mathbf{I}) \propto \exp(-E^{\mathcal{O}}(\mathbf{x}))$. Even if not made explicit, energy function in equation 2.1 and the terms in it depends on current image. Although it is generally computationally infeasible to calculate $P(\mathbf{x}|\mathbf{I})$ exactly due to the partition function, various approximate methods for inference exist, such as approximate *maximum a posteriori* methods (*e.g.* graph-cuts) which minimize Eq. 2.1, or variational methods, such as mean-field approximate $P(\mathbf{x}|\mathbf{I})$ [110], which allow us to approximately estimate a *maximum posteriori marginals* solution (MPM), $x_i^\star = \arg\max_l \sum_{\{\mathbf{x}'|x_i=l\}} P(\mathbf{x}'|\mathbf{I})$.

Typical graph topologies for object class segmentation consider $\mathcal{V}$ to correspond to the pixels of an image, and $\mathcal{E}$ as a 4 or 8-connected neighborhood relation. Recently, mean-field inference methods have also made it possible to use a fully connected graph, where $\mathcal{E}$ connects every pair of pixels, i.e. $\mathcal{E} = \{(i, j)|i, j \in \mathcal{V}, i \neq j\}$ (see [110]) given certain

forms of pairwise potential, and we shall follow this approach in our models. Further, a hierarchical topology may be used, as in [117], which is discussed below.

We set $\psi_i^{\mathcal{O}}(x_i) = -\log(\Pr(X_i = x_i))$, where the probability is derived from a discriminatively trained pixel classifier, TextonBoost [116, 192] [1]. The potential $\psi_{ij}^{\mathcal{O}}(x_i, x_j)$ takes

[1] TextonBoost in this paper means the unary potential in ALE library.

Table 2.1: List of notations

Symbols	Explanation (use RV to represent random variable)
$\mathcal{X}$	Set of RV for object labels: $\mathcal{X} = \{X_1, X_2, ..., X_N\}$
$\mathcal{O}$	Set of object labels: $\mathcal{O} = \{l_1, l_2, ..., l_K\}$
$E^{\mathcal{O}}(\mathbf{x})$	Energy function for segmenting objects
$\psi_i^{\mathcal{O}}$	Unary potential function for object labels
$P(\mathbf{x}\|\mathbf{I})$	The probability of a configuration $\mathbf{x}$ given the observation image $\mathbf{I}$
$\psi_{ij}^{\mathcal{O}}$	Pairwise potential function for object labels
$G(\mathcal{V}, \mathcal{E})$	a graph with vertex $\mathcal{V}$ and connection $\mathcal{E}$
$\mathcal{Y}$	Set of RV for attribute labels: $\mathcal{Y} = \{Y_1, Y_2, ..., Y_N\}$
$\mathcal{A}$	Set of attribute labels: $\mathcal{A} = \{a_1, a_2, ..., a_M\}$
$\mathcal{P}(\mathcal{A})$	Power set of $\mathcal{A}$: $\mathcal{P}(\mathcal{A}) = \{\{\}, \{a_1\}, ..., \{a_1, ..., a_M\}\}$
$E^{\mathcal{A}}(\mathbf{y})$	Energy function for segmenting attributes
X_i	A RV for object label of pixel $i \in \{1, 2, ..., N\}$
$Y_{i,a}$	A RV for attribute $a \in \mathcal{A}$ of pixel i
Y_i	A RV for a set of attributes $\{a : Y_{i,a} = 1\}$ of pixel i
Z_i	A RV $Z_i = (X_i, Y_i)$ of pixel i
$\mathcal{Z}$	RVs of CRF: $\mathcal{Z} = \{Z_1, Z_2, ..., Z_N\}$
$\mathcal{J}$	Joint label set $\mathcal{J} = \mathcal{O} \times \mathcal{P}(\mathcal{A})$
$E^{\mathcal{J}}(\mathbf{x})$	Energy function for joint segmenting objects and attributes
$y_{i,a}, y_i$	Assignment of RVs $Y_i, Y_{i,a}$: $y_{i,a} \in \{0, 1\}, y_i \in \mathcal{P}(\mathcal{A})$
x_i, z_i	Assignment of RVs X_i, Z_i: $x_i \in \mathcal{O}$, $z_i = (x_i, y_i)$
ψ_i	Unary cost of CRF
ψ_{ij}	Pairwise cost of CRF
$\psi_i^{\mathcal{O}}(x_i)$	Cost of X_i taking value $x_i \in \mathcal{O}$
$\psi_{i,a}^{\mathcal{A}}(y_{i,a})$	Cost of $Y_{i,a}$ taking value $y_{i,a} \in \{0, 1\}$
$\psi_{i,o,a}^{\mathcal{O}\mathcal{A}}$	Cost of conflicts between correlated attributes and objects
$\psi_{i,a,a'}^{\mathcal{A}}$	Cost of correlated attributes taking distinct indicators
$\psi_{ij}^{\mathcal{O}}$	Cost of similar pixels with distinct object labels
$E^{\mathcal{J}}(\mathbf{z})$	Energy function for two-level Hierarchical model
$D = \{(\mathbf{f}_1, \hat{\mathbf{z}}_1), ..., (\mathbf{f}_N, \hat{\mathbf{z}}_N)\}$	training data
$H_{t,a}$	Boosting classifier for t round and a attribute
$R(a_1, a_2)$	Correlation between the attribute a_1 and the attribute a_2

the form of a Potts model:

$$\psi_{ij}^{\mathcal{O}}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j, \\ g(i,j) & \text{otherwise.} \end{cases} \tag{2.2}$$

For a fully connected graph topology as in [110] $g(i,j)$ is defined as:

$$\begin{aligned} g(i,j) =& w^{(1)} \exp(-\frac{|p_i - p_j|^2}{2\theta_\mu^2} - \frac{|I_i - I_j|^2}{2\theta_\nu^2}) \\ &+ w^{(2)} \exp(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}), \end{aligned} \tag{2.3}$$

where p_i indicates the location of the ith pixel, I_i indicates the intensity of the ith pixel, and θ_μ,θ_ν, and θ_γ are the parameters.

2.2.2 Multi-label CRF for Attributes

We define a *multi-label* CRF for attributes similarly to the multi-class CRF above, but where the random variables take sets of labels instead of single labels. These sets represent the set of attributes present in a pixel. Formally, we have a set of random variables $\mathcal{Y} = \{Y_1, Y_2, ..., Y_N\}$, and a set of *attribute labels*, $\mathcal{A} = \{a_1, a_2, ..., a_M\}$. Rather than taking values directly in $\mathcal{A}$ though, the Y_i's take values in the *power-set* of the attributes, *i. e.* $y_i \in \mathcal{P}(\mathcal{A})$, where $\mathcal{P}$ is the power-set operator. As in the multi-class case, $\mathbf{y}$ is a joint assignment of these random variables. If we ignore the object labels for now, we can define a multi-label CRF distribution by an energy over $\mathcal{Y}$ as:

$$E^{\mathcal{A}}(\mathbf{y}) = \sum_{i \in \mathcal{V}} \psi_i^{\mathcal{A}}(y_i) + \sum_{\{i,j\} \in \mathcal{E}} \psi_{ij}^{\mathcal{A}}(y_i, y_j), \tag{2.4}$$

and we imply that $P(\mathbf{y}|\mathbf{I}) \propto \exp(-E^{\mathcal{A}}(\mathbf{y}))$. In general, since $|\mathcal{P}(\mathcal{A})|$ grows exponentially with $|\mathcal{A}|$, the number of parameters in $\psi_i^{\mathcal{A}}$ and $\psi_{ij}^{\mathcal{A}}$ will also grow exponentially if we allow arbitrary potential forms. Below, we describe how we factorize these terms, leading to a tractable model at inference time.

We express $\psi_i^{\mathcal{A}}(y_i)$ as follows:

$$\psi_i^{\mathcal{A}}(y_i) = \sum_a \psi_{i,a}^{\mathcal{A}}(y_{i,a}) + \sum_{a_1 \neq a_2} \psi_{i,a_1,a_2}^{\mathcal{A}}(y_{i,a_1}, y_{i,a_2}). \tag{2.5}$$

Here we use auxiliary binary indicator variables $y_{i,a}$, where $y_{i,a} = [a \in y_i]$ (where $[.]$ is the Iverson bracket), which is 1 for a true condition and 0 otherwise (i.e. $y_{i,a}$ indicates whether attribute a is present in the set at pixel i). We set $\psi_{i,a}^{\mathcal{A}}(y_{i,a})$ based on the output of a probabilistic classifier, $\psi_{i,a}^{\mathcal{A}}(b) = -\log(\Pr(y_{i,a} = b))$, $b \in \{0, 1\}$. For this purpose, we

train m independent binary TextonBoost classifiers [116], one for each attribute. Further, we set:

$$\psi^{\mathcal{A}}_{i,a_1,a_2}(y_{i,a_1}, y_{i,a_2}) = \begin{cases} 0 & \text{if } y_{i,a_1} = y_{i,a_2}, \\ R^{\mathcal{A}}(a_1, a_2) & \text{otherwise,} \end{cases} \tag{2.6}$$

where $R^{\mathcal{A}}(a_1, a_2) \in [-1, 1]$ is a learnt *correlation* between a_1 and a_2. Hence, for highly correlated attributes, we pay a high cost if their indicators do not match. We discuss how to learn $R^{\mathcal{A}}$ in Sec. 2.3.

We define $\psi^{\mathcal{A}}_{i,j}(y_i, y_j)$ as follows:

$$\psi^{\mathcal{A}}_{i,j}(y_i, y_j) = \sum_a \psi^{\mathcal{A}}_{i,j,a}(y_{i,a}, y_{j,a}). \tag{2.7}$$

Here, we define $\psi^{\mathcal{A}}_{i,j,a}$ as a Potts model over binary indicators:

$$\psi^{\mathcal{A}}_{i,j,a}(y_{i,a}, y_{j,a}) = \begin{cases} 0 & \text{if } y_{i,a} = y_{j,a}, \\ g(i,j) & \text{otherwise,} \end{cases} \tag{2.8}$$

where, as above, we take $g(i, j)$ as in Eq.2.3 for the fully connected model, allowing us to use filter-based inference.

2.2.3 Factorial CRF for Objects and Attributes

We now describe our combined CRF model for objects and attributes. We define the CRF over random variables $\mathcal{Z} = \{Z_1, Z_2, ..., Z_n\}$, where we take $Z_i = (X_i, Y_i)$, i.e. a combination of an object label and an attribute set. Hence, $z_i \in \mathcal{J} = \mathcal{O} \times \mathcal{P}(\mathcal{A})$, where we write $\mathcal{J}$ for joint label set. We then define a joint CRF in terms of a pairwise energy over the Z_i's as above:

$$E^{\mathcal{J}}(\mathbf{z}) = \sum_{i \in \mathcal{V}} \psi^{\mathcal{J}}_i(z_i) + \sum_{\{i,j\} \in \mathcal{E}} \psi^{\mathcal{J}}_{ij}(z_i, z_j), \tag{2.9}$$

and let $P(\mathbf{z}|\mathbf{I}) \propto \exp(-E^{\mathcal{J}}(\mathbf{z}))$.

Note that, equivalently, we could think of Eq. 2.9 as defining a single multi-label CRF over both object and attribute label sets, i.e. $z_i \in \mathcal{P}(\mathcal{O} \cup \mathcal{A})$. The factorization into multi-class object and multi-label attribute components makes the assumption that any configuration $\mathbf{z}$ has infinite energy (or zero probability) for some i and object labels $l_1 \neq l_2$, $l_1 \in z_i$ and $l_2 \in z_i$, or $l \notin z_i$ for all l. Indeed, it may be appropriate in certain cases to allow multiple object labels at each pixel, for instance if we have a semantic hierarchy including labels such as animal, mammal, dog etc., or a hierarchy of parts such as bicycle, wheel, spoke etc. In this case we would form a product of two multi-label CRF.

We define the joint unary potential as follows:

$$\psi^{\mathcal{J}}_i(z_i) = \psi^{\mathcal{O}}_i(x_i) + \psi^{\mathcal{A}}_i(y_i) + \sum_{l,a} \psi^{\mathcal{O}\mathcal{A}}_{i,l,a}(x_i, y_{i,a}), \tag{2.10}$$

where $\psi_i^{\mathcal{O}}$ and $\psi_i^{\mathcal{A}}$ are defined as above, and the final term takes the form:

$$\psi_{i,l,a}^{\mathcal{OA}}(x_i, y_{i,a}) = \begin{cases} 0 & \text{if } y_{i,a} = [x_i = l] \\ R^{\mathcal{OA}}(l, a) & \text{otherwise,} \end{cases} \tag{2.11}$$

where, as before $R^{\mathcal{OA}}(l, a) \in [-1, 1]$ is a learnt *correlation* between l and a. The first condition in Eq. 2.11 is satisfied if $x_i = l$ holds, and $y_{i,a} = 1$ is also satisfied.

Our joint pairwise term simply combines the individual object and attribute pairwise terms:

$$\psi_{ij}^{\mathcal{J}}(z_i, z_j) = \psi_{ij}^{\mathcal{O}}(x_i, x_j) + \psi_{ij}^{\mathcal{A}}(y_i, y_j). \tag{2.12}$$

2.2.4 Hierarchical Model

In addition to a fully connected CRF over a pixel variable set, we also consider a two-level hierarchial model, where, in addition to labelling object classes and attributes at the *pixel* level, we also label objects and attributes at a *region* level, as shown in Fig. 2.2. We thus consider that our vertex set is partitioned into disjoint sets $\mathcal{V}_{\text{pix}}$ and $\mathcal{V}_{\text{reg}}$, each associated with its own set of attributes, $\mathcal{A}_{\text{pix}}$, $\mathcal{A}_{\text{reg}}$. We maintain dense connectivity over all variables at the pixel level, i.e. $(i, j) \in \mathcal{E}$ for all $i \neq j$ and $i, j \in \mathcal{V}_{\text{pix}}$. For each $j \in \mathcal{V}_{\text{reg}}$, we assume that we have a subset of pixels $\mathcal{S}_j \subset \mathcal{V}_{\text{pix}}$ (which represent the region), and that the edge set contains an edge joining each region variable to all the pixels in its subset, $(i, j) \in \mathcal{E}$ for all $i \in \mathcal{S}_j$. This gives rise to the energy:

$$\begin{aligned} E^{\mathcal{H}}(\mathbf{z}) \quad = \quad & \sum_{i \in \mathcal{V}_{\text{pix}}} \psi_i^{\mathcal{J}}(z_i) + \sum_{\substack{(i,j) \in \mathcal{E}, \\ i,j \in \mathcal{V}_{\text{pix}}}} \psi_{ij}^{\mathcal{J}}(z_i, z_j) \\ & + \sum_{i \in \mathcal{V}_{\text{reg}}} \psi_i^{\mathcal{J}'}(z_i) + \sum_{\substack{(i,j) \in \mathcal{E}, \\ i \in \mathcal{V}_{\text{pix}}, j \in \mathcal{V}_{\text{reg}}}} \psi_{ij}^{\mathcal{J}'}(z_i, z_j), \end{aligned} \tag{2.13}$$

where we implicitly take $\psi_i^{\mathcal{J}}(z_i) = \infty$ if $a \in y_i$ with $i \in \mathcal{V}_{\text{pix}}$ and $a \in \mathcal{A}_{\text{reg}}$, and vice versa for region variables and object attributes.

Similar to [117], we use the efficient object detector [74, 47] and binary segmentation methods [46] to get regions $\mathcal{S}_j$. We thus assume that we have a proposed object class for each region, $o_j \in \mathcal{O}$, $j \in \mathcal{V}_{\text{reg}}$, and an associated score from the detector, s_j. Also, we train a classifier to produce probabilistic outputs for all attributes $\mathcal{A}_{\text{reg}}$ at the region level, and estimate a correlation matrix $R^{\mathcal{OA}_{\text{reg}}}$ between objects and region level attributes. The joint unary terms of a region $\psi_i^{\mathcal{J}'}(z_i)$ then take the same form as Eq. 2.10, except that we set $\psi_i^{\mathcal{O}}(x_i) = 0$ for all x_i, and $\psi_{i,l,a}^{\mathcal{OA}_{\text{reg}}}(x_i, y_{i,a}) = 0$ for all $x_i \neq o_i$. Our region-pixel pairwise terms take the form:

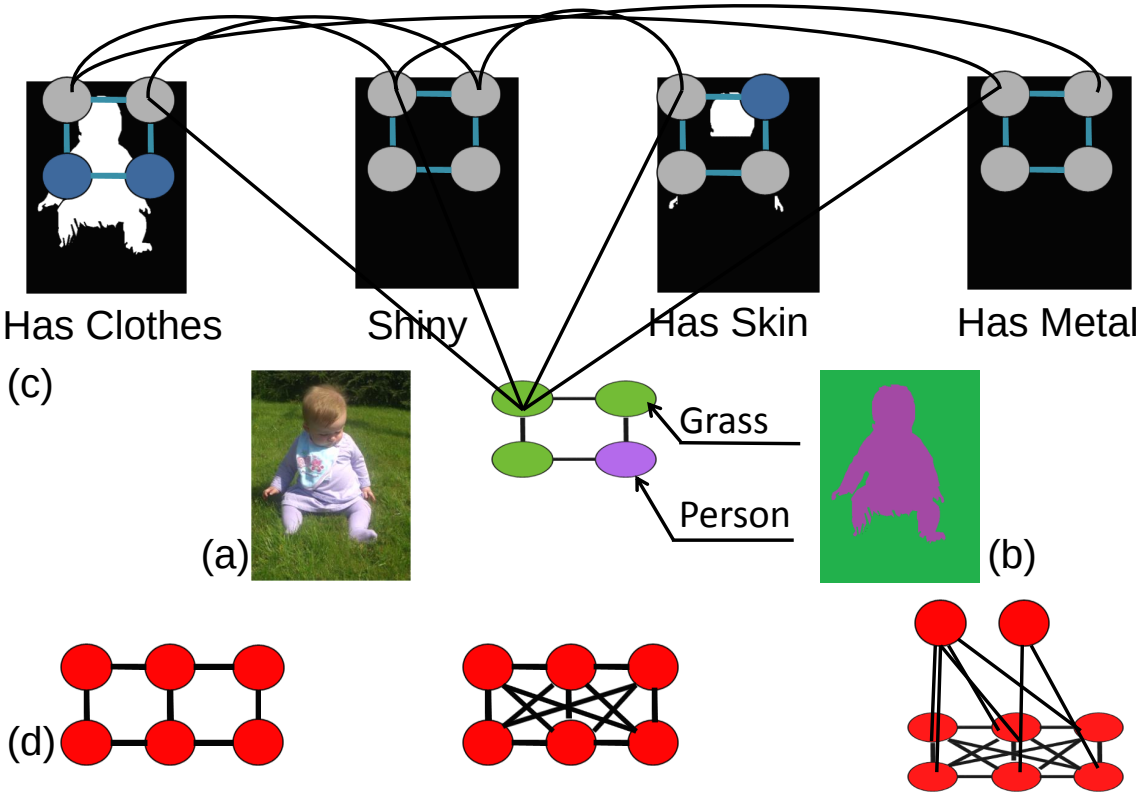


Figure 2.2: **Illustration of Factorial-CRF-based Semantic Segmentation for object classes and Attributes**. (a) shows the input image. (b) shows the ground truth mask image for object classes. (c) shows the attributes masks. (d) compares various CRF topologies including a grid CRF, a fully-connected CRF, and a hierarchial fully connected CRF. Best view in colour.

$$\psi_{ij}^{\mathcal{J}'}(z_i, z_j) = \begin{cases} -s_j & \text{if } x_i = o_j \text{ and } x_j = o_j \\ 0 & \text{otherwise.} \end{cases} \tag{2.14}$$

where, s_j is the score from the jth region associated object detector.

2.2.5 Inference

Following Krähenbühl *et al.* [110], we adopt a mean field approximation approach for inference. This involves finding a mean field approximation $Q(\mathbf{z})$ that minimizes the KL-divergence $D(Q||P)$ among all distributions Q that can be expressed as a product of independent marginals, $Q(\mathbf{z}) = \prod_i Q_i(z_i)$. Given the form of our factorial model, we can factorize Q further into a product of marginals over multi-class object and binary attribute variables. Hence we take $Q_i(z_i) = Q_i^{\mathcal{O}}(x_i) \prod_a Q_{i,a}^{\mathcal{A}}(y_{i,a})$, where $Q_i^{\mathcal{O}}$ is a multi-class distribution over the object labels, and $Q_{i,a}^{\mathcal{A}}$ is a binary distribution over $\{0, 1\}$.

Given this factorization, we can express the required mean field updates (see [109]) for the non-hierarchical model as:

$$\begin{aligned} Q_i^{\mathcal{O}}(x_i = l) &= \frac{1}{Z_i^{\mathcal{O}}} \exp\{-\psi_i^{\mathcal{O}}(l) \\ &\quad - \sum_{j \neq i} Q_j^{\mathcal{O}}(x_j = l)(-g(i,j)) \\ &\quad - \sum_{a,b \in \{0,1\}} Q_{ja}^{\mathcal{A}}(y_{ja} = b)\psi_{i,x_i,a}^{\mathcal{OA}}(l,b)\}, \end{aligned} \tag{2.15}$$

and

$$\begin{aligned} Q_{i,a}^{\mathcal{A}}(y_{i,a} = b) &= \frac{1}{Z_{ia}^{\mathcal{A}}} \exp\{-\psi_{ia}^{\mathcal{A}}(b) \\ &\quad - \sum_{j \neq i} Q_{ja}^{\mathcal{A}}(y_{ja} = b)(-g(i,j)) - \\ &\quad \sum_{a' \neq a, b' \in \{0,1\}} Q_{ia'}^{\mathcal{A}}(y_{ia'} = b')\psi_{i,a,a'}^{\mathcal{A}}(b,b') \\ &\quad - \sum_{l} Q_i^{\mathcal{O}}(x_i = l)\psi_{i,l,a}^{OA}(l,b), \end{aligned} \tag{2.16}$$

where $Z_i^{\mathcal{O}}$ and $Z_{ia}^{\mathcal{A}}$ are per-pixel normalisation factors, and $b \in \{0,1\}$. As in [110], we can efficiently evaluate the pairwise summations in Eq. 2.15 and Eq. 2.16 using $N + M$ Gaussian convolutions given that our pairwise factors take Potts forms as described. Updates for the hierarchical model take a similar form.

2.2.6 Learning parameters for the CRF

For the low-level feature descriptors (LBP, SIFT, HOG, Texton, Colour SIFT), we fixed the parameters for the datasets according to the setting for the best results on PASCAL VOC 2010 dataset using AHCRF [116]. These parameters are tuned based on cross-validation. In this work, we have a two-stages approach. We have used these hand-craft features and the boosting classifiers [192] to obtain the unary potential functions. Then we have the fully-connected CRFs as post-processing step. The detail implementation an be found in ALE library [2]. Regarding the parameters of the CRFs, we use cross-validation [107, 192] to learn the weights for the objects unary responses, attributes unary responses, pairwise, and region-level responses.

[2] http://www.robots.ox.ac.uk/~phst/ale.htm

2.3 Label Correlation Discovery

In this section, we describe a piecewise method for training the label correlation matrices, $R^{\mathcal{A}}$, $R^{\mathcal{OA}}$ and $R^{\mathcal{OA}_{\text{reg}}}$ in the model described. We train all matrices simultaneously by learning an $(N+M)^2$ correlation strength matrix (hence treating the problem as a purely multi-label problem) and then extracting the relevant sub-matrices.

Specifically, we use the modified Adaboost framework of [183, 211] with multiple hypothesis reuse as described in [187]. In training, we denote by $\mathcal{D} = \{(\mathbf{f}_1, \bar{\mathbf{z}}_1), \ldots, (\mathbf{f}_N, \bar{\mathbf{z}}_N)\}$ a training dataset of N instances (i.e. pixels or regions), where $\mathbf{f}_i$ is a feature vector for the i-th instance derived from the image $\mathbf{I}$ (*e.g.* a bag of words vector) and $\bar{\mathbf{z}}_i = [\bar{\mathbf{x}}_i; \bar{\mathbf{y}}_i]$ is an indicator vector of length $N+M$, where $\bar{\mathbf{x}}_i(l) = 1$ implies object l is associated with instance i, and $\bar{\mathbf{x}}_i(l) = -1$ implies it is not, and similarly for $\bar{\mathbf{y}}_i(a) = 1$ for attribute a. $\bar{\mathbf{z}}_i$ is thus a vector representation of a set of objects/attributes present at i.

In the description below, we focus on deriving the attribute-attribute correlations, but the same approach is used for deriving object-attribute correlations. The boosting approach of [187] generates strong classifiers $H_{t,a}(\mathbf{f})$ for each attribute a and each round of boosting, $t = 1...T$. These strong classifiers have the form:

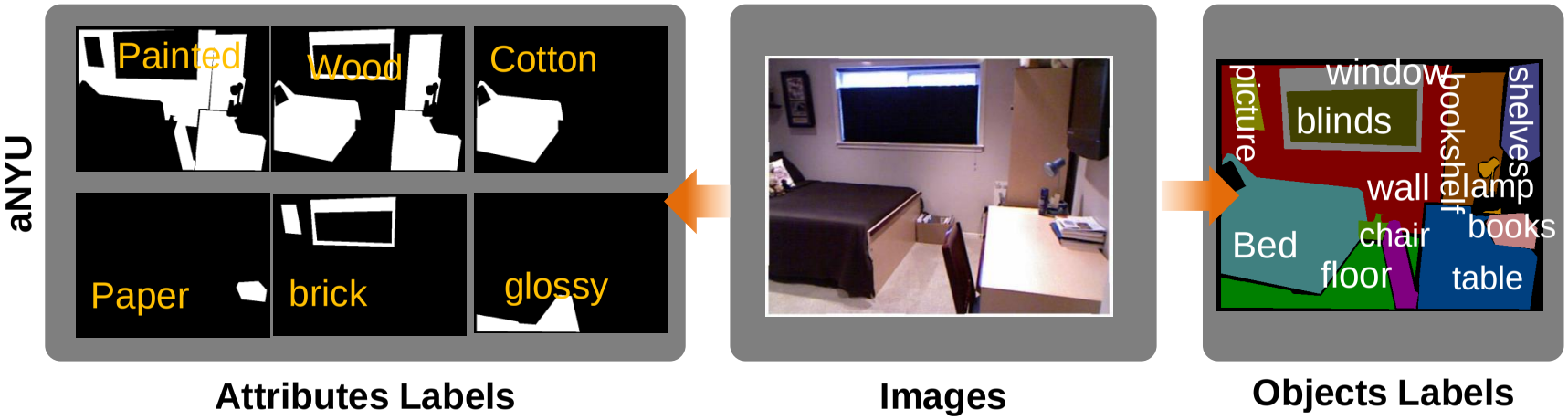
$$H_{t,a} = \sum_{t=1,\ldots,T} \alpha_{t,a} h_{t,a}(\mathbf{f}), \tag{2.17}$$

where $h_{t,a}$ are weak classifiers, and $\alpha_{t,a}$ are the non-negative weights set by the boosting algorithm. Further, the joint learning approach of [187] generates a sequence of *reuse weights* $\beta_{t,a_1}(H_{t-1,a_2})$ for each pair of attributes a_1, a_2 at each iteration t. These represent the weight given to the strong classifier for attribute a_2 in round $t-1$ in the classifier for a_1 at round t. Further, [187] show how these quantities can be used to estimate the label correlation by calculating:

$$\begin{aligned} R(a_1, a_2) \;=\; & \sum_{t=2...T} \alpha_{t,a_1} (\beta_{t,a_1}(H_{t-1,a_2}) \\ & -\beta_{t,a_1}(-H_{t-1,a_2})). \end{aligned} \tag{2.18}$$

Learning the correlations this way incorporates both information about visual appearance similarities and co-occurrence relationships between attributes and objects.

2.4 Datasets



Dataset	Object Labels		Pixel-level Labels		Region-level Labels	
	Number	Names	Number	Names	Number	Names
aNYU	15	Wall, Floor, Picture, Cabinet,...	8	Wood, Painted, Cotton, Glass,...	8	Wood, Painted, Cotton, Glass,...
CORE	27	Airplane, Alligator, Bat,...	9	Bare Metal, Feathers,...	9	Bare Metal, Feathers,..
aPASCAL	20	Aeroplane, Person, Bird, Cat,...	8	Skin, Metal, Plastic, Wood,...	64	2DBoxes, Round, Occluded,...

Figure 2.3: **Annotation illustration.** Extra annotation example and statistics on aNYU, CORE, and aPASCAL datasets. Best view in colour.

We evaluate our approach using three datasets: the Attribute Pascal (aPASCAL) dataset [72], the Cross-category Object REcognition (CORE) dataset [71], and the NYU indoor V2 dataset [193]. In this paper we only use the RGB images from the NYU dataset.

aNYU Dataset. Our first set of experiments is on the RGB images from the NYU V2 dataset [193]. As shown in Fig 2.3, we added 8 additional attribute labels, *i. e. Wood, Painted, Cotton, Glass, Glossy, Plastic, Shiny, and Textured*. We asked 3 annotators to assign material, surface property attributes on each segmentation ground truth region. We then adopted the majority votes from 3 workers as our 8 additional attribute labels. We call this extended dataset the attribute NYU (aNYU) dataset. This dataset has 1449 images collected from 28 different indoor scenes. In our experiments, we select 15 object classes and 8 attributes that have sufficient numbers of instances to train the unary potential. Further, we randomly split the dataset, into 725 images for the training set, 100 for the validation set, and 624 for the testing set.

CORE Dataset. Our second set of experiments is conducted on the Cross-Category Object Recognition (CORE) dataset [71]. This dataset comes with 1049 images and ground truth segmentations for 27 object classes and 9 material attributes. The "objects" set has 27 labels, of which 14 are animals and 13 are vehicles. The "material" set contains nine different materials. Other images in the original CORE dataset are not used because they contain no pixel-level labels. In our experiments, we use 467 images to form the training set, and the remaining 582 images to form a test set. In the original CORE dataset experimental setting [71], some object classes have no training samples. Hence, we move some instances of those objects from test set to the training set.

aPASCAL Dataset. The existing aPASCAL dataset [72] is designed for bounding box level attributes. We transfer the existing 64 bounding-box-level attribute labels to our region-level attributes by finding the closest region segments from the image segmentation ground truth. We select 8 material attributes from 64 as pixel-level attributes, as other attributes are not well-defined on the pixel-level. Among the images in aPASCAL dataset, there are 517 having segmentation ground-truth annotation for both object classes and attributes. We use 191 for testing, and 326 for training.

2.5 Experiments

Our approach is a hierarchical fully-connected CRF model (HI). We compare our approach against the other state-of-the-art image segmentation approaches, including per pixel TextonBoost unary potential [116, 192] (Texton), Pairwise CRF semantic image segmentation

approach (AHCRF [116]), Fully-connected CRF with detection and super-pixel higher orders (Full-C [110, 223]), and Joint attributes-objects Pixel-level fully-connected CRF (JP). JP has the same setting with the proposed approach, but the region-level terms are disabled. The problem of semantic image segmentation for attributes is a multi-label problem and these methods are not designed for dealing with it, so we we treat each as a binary one-vs-all label problem, with no pairwise terms between them, in contrast to our method in which we learn the important correlations between attributes. We also conduct experiments to understand the effect of each term in the proposed full model.

We choose the average intersection/union score as the evaluation measure. This measure is adopted from VOC [68], defined as TP / (TP + FP + FN). TP represents the true positive, and FP means false positive, and FN indicates the false negative. We compute the average intersection/union score across the attribute classes via summing up the intersection/union score for all the binary attribute segmentations and then dividing by the number of attributes.

We have conducted comprehensive evaluation on three datasets including aNYU, CORE, and aPASCAL. Compared with 5 other methods, we observe that HI outperforms the other approaches across all datasets, as illustrated in Fig. 2.4. In Fig. 2.4, HI achieves higher performance than JP, indicating that exchanging information between attributes and objects at both levels helps to predict both types of variable. Moreover, we observe a significant qualitative improvement, and we believe that a higher percentage increase would be archived if the datasets had more finely labelled data in the test set.

Dataset	Average intersect-union[68](%)											
	Object segment						Attribute segment					
	Texton [116, 192]	AHCRF [116]	Full-C [110, 223]	JP	HI		Texton [116, 192]	AHCRF [116]	Full-C [110, 223]	JP	HI	
aNYU	17.4	20.0	18.9	20.8	**22.0**		8.90	10.1	10.0	14.4	**15.1**	
CORE	17.5	17.6	17.5	19.1	**20.1**		15.6	17.0	17.4	17.5	**17.8**	
aPASCAL	27.0	30.3	36.9	36.4	**37.1**		15.0	16.5	16.5	16.9	**17.6**	

Figure 2.4: **Quantitative and quantitative results.** Results on the aNYU, CORE [71] and aPASCAL [72] datasets. We compare 5 different approaches: TextonBoost(Texton [116, 192]), Pairwise CRF with detection and super-pixel higher orders (AHCRF [116]), Fully-connected CRF with detection and super-pixel higher orders (Full-C [110, 223]), Joint Pixel-level CRF (JP), and Hierarchical CRF (HI). The results are reported as average intersection-union [68]. We obtain the attribute unary potentials with multiple binary segmentation, using the AHCRF [116] library. The attribute segmentation results for the method Full-C are obtained using Dense CRF inference based on these attribute unary potentials. Best view in colour.

Effect of attribute terms. To clarify the effect of each attribute term in Eq. 2.13, we report the performance of object segmentation, using HI with different components being disabled. We take the learned models and remove, in turn, each type of attribute term (i.e. the joint attributes-objects term, the joint attributes-attributes term, the attributes in region level, and the attributes in pixel level), and report the performance in Table 2.2. When we remove the per-pixel attribute assignment, the object segmentation accuracy reduces by 5%, but when we remove the region-level attributes, the accuracy reduces by 4.4%. This suggests per-pixel attribute assignment is important to achieve higher accuracy and finer segmentation.

	Average label-accuracy(%) for object segmentation			
Dataset	full model	W/O pix-att	W/O region-att	W/O att
aNYU	61.4	56.4	57.0	51.3

Table 2.2: **Effect of different terms in our model.** We compare the average object label-accuracy(%) of our full model without (W/O) different components. "Full model" means the proposed approach, the hierarchical semantic image segmentation for both objects and attributes. "W/O pix-att" indicates the one without pixel-level attribute terms, "W/O region-att" represents the one without region-level attribute terms, and "W/O att" is the one without attribute terms.

In addition, to understand the potential of using attributes in helping semantic image segmentation, we evaluate the performance improvement of HI by setting the attribute factors to the ground truth labels (as if we had a perfect attribute CRF). Result shows 42% average label accuracy improvement on the object class segmentation, compared against the results of the proposed joint inference approach. This suggests that there is still great potential in using attributes towards semantic image segmentation.

Joint Inference Timings. All the experiments are carried out on a machine with a Intel Xeon $\text{E5} - 2687\text{W}$ (3.1GHz, 1600MHz) and 64.0GB. For the hierarchical model, the straightforward implementation of the inference takes on average 11 seconds per image on the aNYU dataset, where the image size is 620×460. This inference can easily be parallelized. By enabling OpenMP and optimizing the implementation, the inference part can achieve 1.2 seconds per 620×460 image, on all 16 cores of the same machine. Further speed boost can be achieved with GPU implementation.

2.6 Conclusions and Future Work

In this chapter, we have proposed a joint approach to simultaneously predict the attribute and object class labels for pixels and regions in a given image. The experiments suggest

that combining information from attributes and objects at region and pixel-levels helps semantic image segmentation for both object classes and attributes. Further experiments also show that per-pixel attribute segmentation is important in achieving higher accuracy and finer semantic segmentation results. In order to encourage future work on the problem of semantic image segmentation with objects and attributes, we expand the aNYU dataset by adding per-pixel attribute annotation.

In future work, we intend to consider allowing multi-label object predictions as well as attributes, and combining our piecewise learning approach to jointly learn all the parameters. We also plan to achieve the GPU implementation for the proposed approach and generalize current approach for 3D scenes understanding. It is possible to extend the set of object and attribute labels and maintain efficiency by following Sturgess *et al.* [197].

With objects and visual attributes, this allows a new way of human-computer interaction. In next chapter, we would like to make use of the objects and visual attributes, and we develop a verbal guided image parsing system.

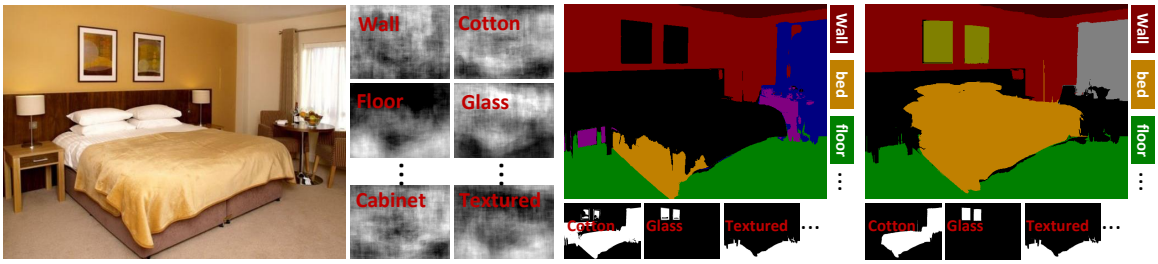
Chapter 3

ImageSpirit: Verbal Guided Image Parsing

Humans describe images using nouns and adjectives while algorithms operate on images represented as sets of pixels. Bridging this gap between how humans would like to access images versus their typical representation is the goal of image parsing, which involves assigning object and attribute labels to a pixel. We introduced the joint segmentation for objects and attributes in chapter 2. In this chapter, we propose treating nouns as object classes and adjectives as visual attributes. This treatment allows us to formulate the image parsing problem as one of jointly estimating per-pixel object and attribute labels from a set of training images. We propose an efficient (interactive time) solution. By using the extracted objects/attributes labels as handles, our system empowers a user to refine the results verbally. This function enables hands-free parsing of an image into pixel-wise object/attribute labels that correspond to human semantics. Verbally selecting objects of interest enables a novel and natural interaction modality that can be used to interact with new generation devices (e.g. smartphones, Google Glass, living room devices). We demonstrate our system on a large number of real-world images with varying complexity. We report the results of both a large-scale quantitative assessment and a user study, to understand the tradeoffs between our system and traditional mouse-based interactions.

3.1 Introduction

Humans describe images in terms of language components such as nouns (e.g. bed, cupboard, desk) and adjectives (e.g. textured, wooden). In contrast, pixels form a natural representation for computers [76]. Bridging this gap between our mental models and machine representation is the goal of image parsing [214, 208]. The goals of this chapter are



(a) Inputs: an image and object/attributes potentials [192]

(b) Automatic scene parsing results

(c) Natural language guided parsing

Figure 3.1: (a) Given a source image downloaded from the Internet, our system generates multiple weak object/attributes cues. (b) Using a novel multi-label CRF, we generate an initial per-pixel object and attribute labeling. (c) The user provides the verbal guidance: 'Refine the cotton bed in center-middle', 'Refine the white bed in center-middle', 'Refine the glass picture', 'Correct the wooden white cabinet in top-right to window' allows reweighting of CRF terms to generate, at interactive rates, high quality scene parsing result.

two-fold: develop a new automatic image parsing model that can handle attributes (adjectives) and objects (nouns), and explore how to interact verbally with this parse in order to improve the results. This is a difficult problem. Whilst to date there exists a large number of automated image parsing techniques [116, 192, 110, 114, 206], their parsing results often require additional refinement before being useful for applications such as image editing. In this chapter, we propose an efficient approach that allows users to produce high quality image parsing results from verbal commands. Such a scheme enables hands-free parsing of an image into pixel-wise object and attribute labels that are meaningful to both humans and computers. The speech (or speech & touch) input is useful for the new generation of devices such as smart phones, Google Glass, consoles and living room devices, which do not readily accommodate mouse interaction. Such an interaction modality not only enriches how we interact with the images, but also provides an important interaction capability for applications where non-touch manipulation is crucial [95] or hands are busy in other ways [92].

We face three technical challenges in developing verbal guided[1] image parsing: (i) words are concepts that are difficult to translate into pixel-level meaning; (ii) how to best update the parse using verbal cues; and (iii) ensuring the system responds at interactive rates. To address the first problem, we treat nouns as objects and adjectives as attributes. Using training data, we obtain a score at each pixel for each object and attribute, e.g. Fig. 3.1(a). These scores are integrated through a novel, multi-label factorial conditional random field (CRF)

[1] We use the term verbal as a short hand to indicate word-based, i.e., nouns, adjectives, and verbs. We make this distinction as we focus on semantic image parsing rather than speech recognition or natural language processing.

model that jointly estimates both object and attribute predictions. This is different from chapter 2 for the overall system speed consideration, since the proposed system in this chapter is an interactive system. In chapter 2, we generalise this model to include hierarchical relations between regions and pixels, improved attribute-object relationship learning, etc. We show how to perform inference on this model to obtain an initial scene parse as demonstrated in Fig. 3.1(b). This joint image parsing with both objects and attributes provides verbal handles on the underlying image which we can now use for further manipulation of the image. Furthermore, our modeling of the symbiotic relation between attributes and objects results in a higher quality parsing than considering each separately [116, 110]. To address the second problem, we show how the user commands can be used to update the terms of the CRF. This process of verbal command updating cost, followed by automatic inference to get the results, is repeated until satisfactory results are achieved. Putting the human in the loop allows one to quickly obtain very good results. This is because the user can intuitively leverage a high level understanding of the current image and quickly find discriminative visual attributes to improve scene parsing. For example, in Fig. 3.1(c), if the verbal command contains the words 'glass picture', our algorithm can reweight the CRF to allow improved parsing of the 'picture' and the 'glass'. Finally, we show that our joint CRF formulation can be factorized. This permits the use of efficient filtering based techniques [110] to perform inference at interactive speed.

We evaluate our approach on the attribute-augmented NYU V2 RGB image dataset [193] that contains 1449 indoor images. We compare our results with state-of-the-art object-based image parsing algorithms [116, 110]. We report a 6% improvement in terms of average label accuracy (ALA) [2] using our automated object/attribute image parsing. Beyond these numbers, our algorithm provides critical verbal handles for refinement and subsequent edits leading to a significant improvement (30% ALA) when verbal interaction is allowed. Empirically, we find that our interactive joint image parsing results are better aligned with human perception than those of previous non-interactive approaches. Further, we find our method performs well on similar scene types taken from outside of our training database. For example, our indoor scene parsing system works on internet images downloaded using 'bedroom' as a search word in Google.

Whilst scene parsing is important in its own right, we believe that our system enables novel human-computer interactions. Specifically, by providing a hands-free selection mechanism to indicate objects of interest to the computer, we can largely replace the role traditionally filled by the mouse. This enables interesting image editing modalities such as verbal guided image manipulation which can be integrated in smart phones and Google

[2] Label accuracy is defined as the number of pixels with correct label divided by the total number of pixels.

Glass, by making commands such as 'zoom in on the cupboard in the far right' meaningful to the computer.

In summary, our main contributions are:

1. a new interaction modality that enables verbal commands to guide image parsing;
2. the development of a novel multi-label factorial CRF that can integrate cues from multiple sources at interactive rates; and
3. a demonstration of the potential of this approach to make conventional mouse-based tasks hands-free.

3.2 Related works

Object class image segmentation and visual attributes: Assigning an object label to each image pixel, known as object class image segmentation or scene parsing, is one of computer vision's core problems. TextonBoost [192] is a ground breaking work for addressing this problem. It simultaneously achieves pixel-level object class recognition and segmentation by jointly modeling patterns of texture and their spatial layout. Several refinements of this method have been proposed, including context information modeling [170], joint optimization of stereo and object label [118], dealing with partial labeling [219], and efficient inference [110]. These methods deal only with object labels (noun) and not attributes (adjectives). Visual attributes [76] and data association [145], which describe important semantic properties of objects, have been shown to be an important factor for improving object recognition [70, 226], scene attributes classification [162], and even modeling of unseen objects [121]. These works have been limited to determining the attributes of an image region contained in a rectangular bounding box. Recently, Tighe and Lazebnik [206] have addressed the problem of parsing image regions with multiple label sets. However, their inference formulation remains unaware of object boundaries and the obtained object labeling usually spreads over the entire image. We would like to tackle the problem of image parsing with both objects and attributes. This is a very difficult problem as, in contrast to traditional image parsing in which only one label is predicted per pixel, there now might be zero, one, or a set of labels predicted for each pixel, e.g. a pixel might belong to wood, brown, cabinet, and shiny. Our model is defined on pixels with fully connected graph topology, which has been shown [110] to be able to produce fine detailed boundaries.

Interactive image labeling: Interactive image labeling is an active research field. This field has two distinct trends. The first involves having some user defined scribbles or

bounding boxes, which are used to assist the computer in cutting out the desired object from image [139, 130, 178, 126]. Gaussian mixture models (GMM) are often employed to model the colour distribution of foreground and background. Final results are achieved via Graph Cut [24]. While widely used, these works do not extend naturally to verbal parsing as the more direct scribbles cannot be replaced with vague verbal descriptions such as 'glass'. The second trend in interactive image labeling incorporates a human-in-the-loop [29, 225], which focuses on recognition of image objects rather than image parsing. They resolve ambiguities by interactively asking users to click on the object parts and answer yes/no questions. Our work can be considered a verbal guided human-in-the-loop image parsing. However, our problem is more difficult than the usual human-in-the-loop problems because of the ambiguity of words (as opposed to binary answers to questions) and the requirement for fine pixel wise labeling (as opposed to categorization). This precludes usage of a simple tree structure for querying and motivates our more sophisticated, interactive joint CRF model to resolve the ambiguities.

Semantic-based region selection: Manipulation in the semantic space [18] is a powerful tool and there are a number of approaches. An example is Photo Clip Art [120] which allows users to directly insert new semantic objects into existing images, by retrieving suitable objects from a database. This work has been further extended to sketch based image composition by automatically extracting and selecting suitable salient object candidates [46] from Internet images [37, 41, 82]. Carroll *et al.* [31] enables perspective aware image warps by using user annotated lines as projective constrains. Cheng *et al.* [45] analyze semantic object regions as well as layer relations according to user input scribble marking, enabling interesting interactions across repeating elements. Zhou *et al.* [245] proposed to reshape human image regions by fitting an appropriate 3D human model. Zheng *et al.* [244] partially recover the 3D of man-made environments, enabling intuitive non-local editing. However, none of these methods attempt interactive verbal guided image parsing which has the added difficulty of enabling the use of verbal commands to provide vague guidance cues.

Speech interface: Speech interfaces are deployed when mouse based interactions are infeasible or cumbersome. Although research on integrating speech interfaces into software started in the 1980s [22], it is only recently that such interfaces have been widely deployed (e.g. Apple's Siri, PixelTone [122]). However, most speech interface research is focused on natural language processing and to our knowledge there has been no prior work addressing image region selection through speech. The speech interface that most resembles our work is PixelTone [122], which allows users to attach object labels to scribble based segments.

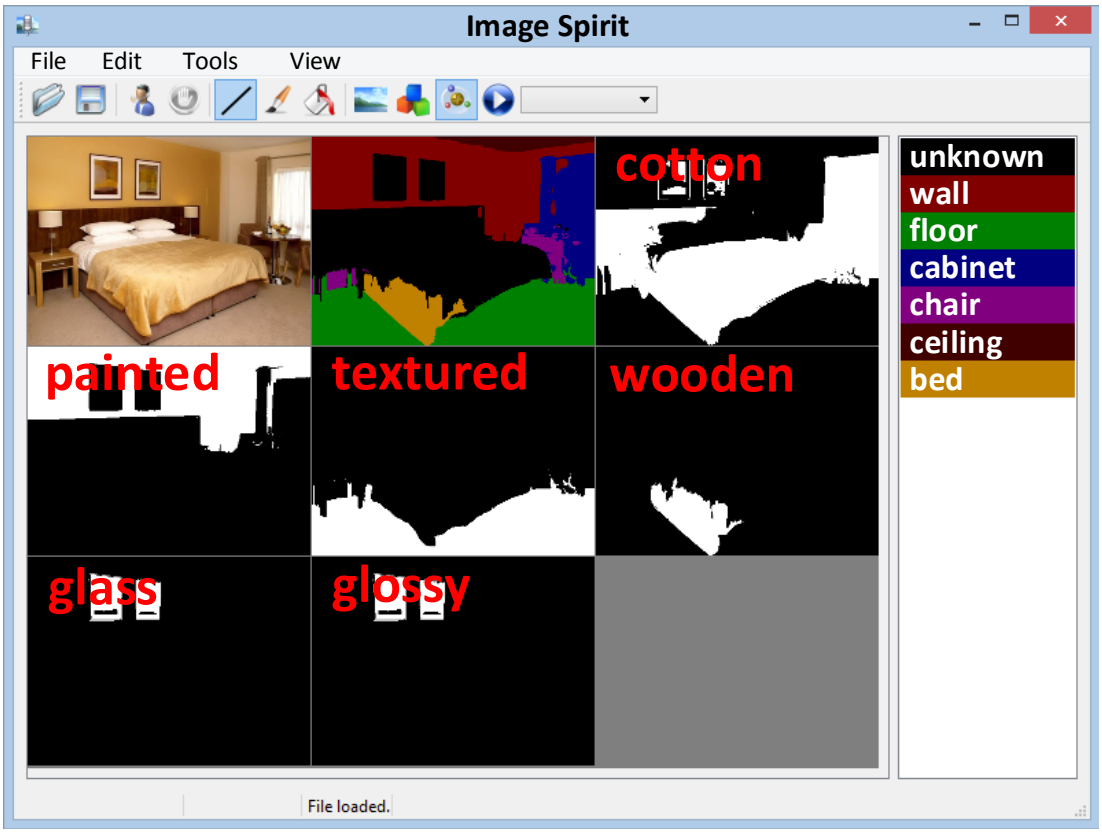


Figure 3.2: User interface of our system (labeling thumbnail view).

These labels allow subsequent voice reference. Independently, we have developed a hands-free parsing of an image into pixel-wise object/attribute labels that correspond to human semantics. This provides a verbal option for selecting objects of interest and is potentially, a powerful additional tool for speech interfaces.

3.3 System Design

Our goal is a verbal guided image parsing system that is simple, fast, and most importantly, intuitive, i.e. allowing an interaction mode similar to our everyday language. After the user loads an image, our system automatically assigns an object class label (noun) and sets of attribute labels (adjectives) to each pixel. Based on the initial automatic image parsing results, our system identifies a subset of objects and attributes that are most related to the image. In Fig. 3.2, to speed up the inference in the verbal refinement stage, our system only considers the subset instead of the whole set of object classes and the attribute labels. The initial automatic image parsing results also provide the bridge between image pixels and verbal commands. Given the parse, the user can use his/her knowledge about

the image to strengthen or weaken various object and attribute classes. For example, the initial results in Fig. 3.2 might prompt the user to realize that the bed is missing from the segmentation but the 'cotton' attribute covers a lot of the same area as is covered by the bed in the image. Thus, the simple command 'Refine the cotton bed in center-middle' will strengthen the association between cotton and bed, allowing a better segmentation of the bed. Note that the final object boundary does not necessarily follow the original boundary of the attribute because verbal information is incorporated only as soft cues, which are interpreted by a CRF within the context of the other information. Algorithm 1 presents a high level summary of our verbal guided image parsing pipeline, with details explained in the rest of this section.

Once objects have been semantically segmented, it becomes straightforward to manipulate them using verb-based commands such as move, change, etc. As a demonstration of this concept, we encapsulate a series of rule-based image processing commands needed to execute an action, allowing hands-free image manipulation (see Section 3.5).

3.3.1 Mathematical Formulation

We formulate simultaneous semantic image parsing for object class and attributes as a multi-label CRF that encodes both object and attribute classes, and their mutual relations. This is a combinatorially large problem. If each pixel takes one of the 16 object labels and a subset of 8 different attribute labels, there are $(16 \times 2^8)^{640\times480}$ possible solutions

Algorithm 1 Verbal guided image parsing.

Input: an image and object/attributes potentials (see Fig. 3.1).
Output: an object and a set of attributes labels for each pixel.
Initialize: object/attributes potentials for each pixel; find pairwise potentials by (3.4).
for Automatic inference iterations i = 1 to T_a **do**
    Update potentials using (3.6) and (3.7) for all pixels simultaneously using efficient filtering technique;
end for
for each verbal input **do**
    update potentials (c.f. Section 3.3.3) according to user input;
    for Verbal interaction iterations i = 1 to T_v **do**
        Update potentials using (3.6) and (3.7) as before;
    end for
end for
Extract results from potentials: at any stage, labels for each pixel could be found by selecting the largest object potential, or comparing the positive and negative attributes potentials.

to consider for an image of resolution 640×480. Direct optimization over such a huge number of variables is computational infeasible without some choice of simplification. The problem becomes more complicated if correlation between attributes and objects are taken into account. In this chapter, we propose using a factorial CRF framework [200] to model correlation between objects and attributes.

A multi-label CRF for dense image parsing of objects and attributes can be defined over random variables $\mathcal{Z}$, where each $Z_i = (X_i, Y_i)$ represents object and attributes variables of the corresponding image pixel i (see Table 3.1 for a list of notations). X_i will take a value from the set of object labels, $x_i \in \mathcal{O}$. Rather than taking values directly in the set of attribute labels $\mathcal{A}$, Y_i takes values from the power-set of the attributes. For example, $y_i = \{wood\}$, $y_i = \{wood, painted, textured\}$, and $y_i = \emptyset$ are all valid assignments. We denote by $\mathbf{z}$ a joint configuration of these random variables, and $\mathbf{I}$ the observed image data. Our CRF model is defined as the sum of per pixel and pair of pixel terms:

$$E(\mathbf{z}) = \sum_i \psi_i(z_i) + \sum_{i<j} \psi_{ij}(z_i, z_j), \tag{3.1}$$

where i and j are pixel indices that range from 1 to N. The per pixel term $\psi_i(z_i)$ measures the cost of assigning an object label and a set of attributes label to pixel i, considering

Table 3.1: List of notations

Symbols	Explanation (use RV to represent random variable)
$\mathcal{O}$	Set of object labels: $\mathcal{O} = \{o_1, o_2, ..., o_K\}$
$\mathcal{A}$	Set of attribute labels: $\mathcal{A} = \{a_1, a_2, ..., a_M\}$
$\mathcal{P}(\mathcal{A})$	Power set of $\mathcal{A}$: $\mathcal{P}(\mathcal{A}) = \{\{\}, \{a_1\}, ..., \{a_1, ..., a_M\}\}$
X_i	A RV for object label of pixel $i \in \{1, 2, ..., N\}$
$Y_{i,a}$	A RV for attribute $a \in \mathcal{A}$ of pixel i
Y_i	A RV for a set of attributes $\{a : Y_{i,a} = 1\}$ of pixel i
Z_i	A RV $Z_i = (X_i, Y_i)$ of pixel i
$\mathcal{Z}$	RVs of CRF: $\mathcal{Z} = \{Z_1, Z_2, ..., Z_N\}$
$y_{i,a}, y_i$	Assignment of RVs $Y_i, Y_{i,a}$: $y_{i,a} \in \{0, 1\}, y_i \in \mathcal{P}(\mathcal{A})$
x_i, z_i	Assignment of RVs X_i, Z_i: $x_i \in \mathcal{O}$, $z_i = (x_i, y_i)$
ψ_i	Unary cost of CRF
ψ_{ij}	Pairwise cost of CRF
$\psi_i^{\mathcal{O}}(x_i)$	Cost of X_i taking value $x_i \in \mathcal{O}$
$\psi_{i,a}^{\mathcal{A}}(y_{i,a})$	Cost of $Y_{i,a}$ taking value $y_{i,a} \in \{0, 1\}$
$\psi_{i,o,a}^{\mathcal{O}\mathcal{A}}$	Cost of conflicts between correlated attributes and objects
$\psi_{i,a,a'}^{\mathcal{A}}$	Cost of correlated attributes taking distinct indicators
$\psi_{ij}^{\mathcal{O}}$	Cost of similar pixels with distinct object labels
$\psi_{i,j,a}^{\mathcal{A}}$	Cost of similar pixels with distinct attribute labels

learned pixel classifiers for both objects and attributes, as well as learned object-attribute and attribute-attribute correlations. The cost term $\psi_{ij}(z_i, z_j)$ encourages similar and nearby pixels to take similar labels.

To optimize (3.1) we break it down into multi-class and binary subproblems using a factorial CRF framework [200], whilst maintaining correlations between object and attributes. The pixel term is decomposed into:

$$\begin{aligned}\psi_i(z_i) &= \psi_i^{\mathcal{O}}(x_i) + \sum_a \psi_{i,a}^{\mathcal{A}}(y_{i,a}) + \sum_{o,a} \psi_{i,o,a}^{\mathcal{OA}}(x_i, y_{i,a}) \\ &\quad + \sum_{a \neq a'} \psi_{i,a,a'}^{\mathcal{A}}(y_{i,a}, y_{i,a'}) \end{aligned} \tag{3.2}$$

where the cost of pixel i taking object label x_i is $\psi_i^{\mathcal{O}}(x_i) = -\log(\Pr(x_i))$, with probability derived from trained pixel classifier (TextonBoost [192]). For each of the M attributes, we train independent binary TextonBoost classifiers, and set $\psi_{i,a}^{\mathcal{A}}(y_{i,a}) = -\log(\Pr(y_{i,a}))$ based on the output of this classifier. Finally, the terms $\psi_{i,o,a}^{\mathcal{OA}}(x_i, y_{i,a})$ and $\psi_{i,a,a'}^{\mathcal{A}}(y_{i,a}, y_{i,a'})$ are the costs of correlated objects and attributes with distinct indicators. They are defined as:

$$\begin{aligned}\psi_{i,o,a}^{\mathcal{OA}}(x_i, y_{i,a}) &= [[x_i = o] \neq y_{i,a}] \cdot \lambda_{\mathcal{OA}} R^{\mathcal{OA}}(o, a) \\ \psi_{i,a,a'}^{\mathcal{A}}(y_{i,a}, y_{i,a'}) &= [y_{i,a} \neq y_{i,a'}] \cdot \lambda_{\mathcal{A}} R^{\mathcal{A}}(a, a') \end{aligned} \tag{3.3}$$

where Iverson bracket, $[.]$, is 1 for a true condition and 0 otherwise, $R^{\mathcal{OA}}(o, a)$ and and $R^{\mathcal{A}}(a, a')$ are derived from learned object-attribute and attribute-attribute correlations respectively. Here $\psi_{i,o,a}^{\mathcal{OA}}(x_i, y_{i,a})$ and $\psi_{i,a,a'}^{\mathcal{A}}(y_{i,a}, y_{i,a'})$ penalize inconsistent object-attributes and attribute-attribute labels by the cost of their correlation value. These correlations are obtained from the phi coefficient (also referred to as the "mean square contingency coefficient [51]), which is learnt from the labeled dataset using [212]. A visual representation of these correlations is given in Fig. 3.3.

The cost term $\psi_{ij}(z_i, z_j)$ can be factorized as object label consistency term and attributes label consistency terms:

$$\psi_{ij}(z_i, z_j) = \psi_{ij}^{\mathcal{O}}(x_i, x_j) + \sum_a \psi_{i,j,a}^{\mathcal{A}}(y_{i,a}, y_{j,a}), \tag{3.4}$$

here we assume each has the form of Potts model [168]:

$$\begin{aligned}\psi_p^{\mathcal{O}}(x_i, x_j) &= [x_i \neq x_j] \cdot g(i, j) \\ \psi_{i,j,a}^{\mathcal{A}}(y_{i,a}, y_{j,a}) &= [y_{i,a} \neq y_{j,a}] \cdot g(i, j).\end{aligned}$$

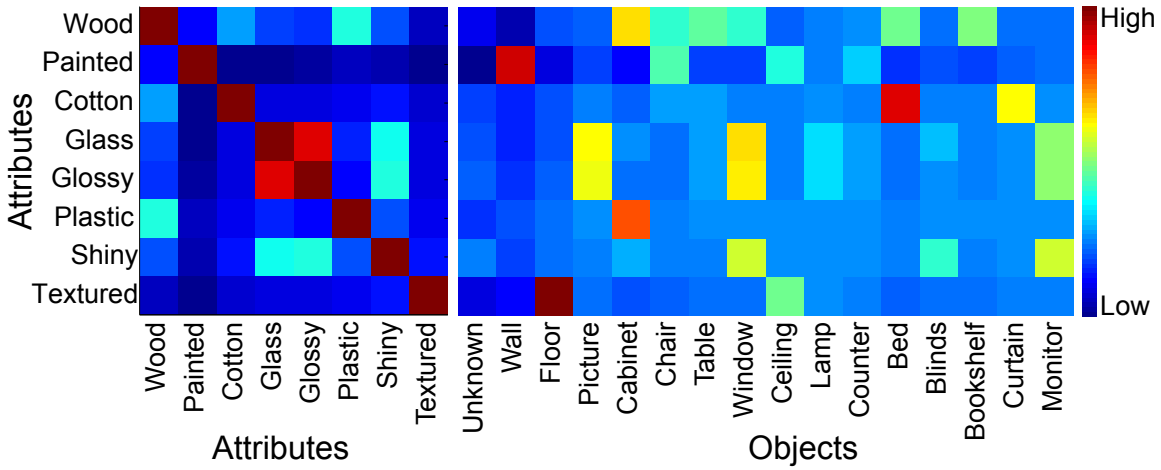


Figure 3.3: Visualization of the $R^{\mathcal{OA}}, R^{\mathcal{AA}}$ terms used to encode object-attribute and attribute-attribute relationships.

We define $g(i,j)$ in terms of similarity between colour vectors I_i, I_j and position values p_i, p_j:

$$\begin{aligned} g(i,j) &= w_1 \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\mu^2} - \frac{|I_i - I_j|^2}{2\theta_\nu^2}\right) \\ &\quad + w_2 \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}\right). \end{aligned} \tag{3.5}$$

All the parameters $\lambda_{\mathcal{OA}}$, $\lambda_{\mathcal{A}}$, w_1, w_2, θ_μ, θ_ν, and θ_γ are learnt via cross validation.

3.3.2 Efficient Joint Inference with Factorized Potentials

To enable continuous user interaction, our system must have a response rate which is close to real time. Recently there has been a breakthrough in the mean-field solution of random fields, based on advances in filtering based methods in computer graphics [3, 110]. Here we briefly sketch how this inference can be extended to multi label CRFs.

This involves finding a mean-field approximation $Q(\mathbf{z})$ of the true distribution $P \propto \exp(-E(z))$, by minimizing the KL-divergence $D(Q||P)$ among all distributions Q that can be expressed as a product of independent marginals, $Q(\mathbf{z}) = \prod_i Q_i(z_i)$. Given the form of our factorial model, we can factorize Q further into a product of marginals over multi-class object and binary attribute variables. Hence we take $Q_i(z_i) = Q_i^{\mathcal{O}}(x_i) \prod_a Q_{i,a}^{\mathcal{A}}(y_{i,a})$, where $Q_i^{\mathcal{O}}$ is a multi-class distribution over the object labels, and $Q_{i,a}^{\mathcal{A}}$ is a binary distribution over $\{0, 1\}$.

Given this factorization, we can express the required mean-field updates (c.f. [109]) as:

$$\begin{aligned} Q_i^{\mathcal{O}}(x_i = o) = & \frac{1}{Z_i^{\mathcal{O}}} \exp\{-\psi_i^{\mathcal{O}}(x_i) \\ & - \sum_{i \neq j} Q_j^{\mathcal{O}}(x_j = o)(-g(i,j)) \\ & - \sum_{a \in \mathcal{A}, b \in \{0,1\}} Q_{i,a}^{\mathcal{A}}(y_{i,a} = b)\psi_{i,o,a}^{\mathcal{O}\mathcal{A}}(o,b)\} \end{aligned} \tag{3.6}$$

$$\begin{aligned} Q_{i,a}^{\mathcal{A}}(y_{i,a} = b) = & \frac{1}{Z_{i,a}^{\mathcal{A}}} \exp\{-\psi_{i,a}^{\mathcal{A}}(y_{i,a}) \\ & - \sum_{i \neq j} Q_{j,a}^{\mathcal{A}}(y_{j,a} = b)(-g(i,j)) \\ & - \sum_{a' \neq a \in \mathcal{A}, b' \in \{0,1\}} Q_{i,a'}^{\mathcal{A}}(y_{i,a'} = b')\psi_{i,a,a'}^{\mathcal{A}}(b,b') \\ & - \sum_{o} Q_i^{\mathcal{O}}(x_i = o)\psi_{i,o,a}^{OA}(o,b)\} \end{aligned} \tag{3.7}$$

where $Z_i^{\mathcal{O}}$ and $Z_{ia}^{\mathcal{A}}$ are per-pixel object and attributes normalisation factors. As shown in (3.6) and (3.7), directly applying these updates for all pixels requires expensive sum operations, whose computational complexity is quadratic in the number of pixels. Given that our pair of pixel terms are of Potts form modulated by a linear combination of Gaussian kernels as described in (3.5), simultaneously finding these sums for all pixels can be achieved at a complexity linear in the number of pixels using efficient filtering techniques [3, 110].

3.3.3 Refine Image Parsing with Verbal Interaction

Since the image parsing results of the automatic approach described in Section 3.3.1 are still far away from what a human can perceive from the image and what is required by most image parsing applications such as photo editing, we introduce a verbal interaction modality so that the user can refine the automatic image parsing results by providing a few verbal commands. Each command will alter one of the potentials given in Section 3.3.1.

Supported object classes (**Obj**) include the 16 keywords in our training object class list (bed, blinds, bookshelf, cabinet, ceiling, chair, counter, curtain, floor, lamp, monitor, picture, table, wall, window and unknown). We also support 4 material attributes (**MA**) keywords (wooden, cotton, glass, plastic) and 4 surface attributes (**SA**) keywords (painted, textured, glossy, shiny). For colour attributes (**CA**), we support the 11 basic colour names, suggested by Linguistic study [16]. These colours names/attributes are: black, blue, brown, grey, green, orange, pink, purple, red, white and yellow. Also as observed by [122], humans

Basic definitions:
MA, **SA**, **CA**, **PA**, are attributes keywords in Section 3.3.3.
Obj is an object class name keyword in Section 3.3.3.
ObjDes := [**CA**] [**SA**] [**MA**] **Obj** [in **PA**]
DeformType ∈ {'lower', 'taller', 'smaller', 'larger'}
MoveType ∈ {'down', 'up', 'left', 'right'}

Verbal commands for image parsing:
Refine the **ObjDes**.
Correct the **ObjDes** as **Obj**.
Verbal commands for manipulation:
Activate the **ObjDes**.
Make the **ObjDes DeformType**.
Move the **ObjDes MoveType**.
Repeat the **ObjDes** and move **MoveType**.
Change the **ObjDes** [from **Material/colour**] to **Material/colours**.

Figure 3.4: Illustration of supported verbal commands for image parsing and manipulation (Section 3.5). The brackets '[]' represent optional words.

are not good at describing precise locations but can easily refer to some rough positions in the image. We currently support 9 rough positional attributes (**PA**), by combining 3 vertical positions (top, center, and bottom) and 3 horizontal positions (left, middle, and right).

Fig. 3.4 illustrates the 7 commands that are currently supported. These command can alter the per pixel terms in (3.2). Notice that both the image parsing commands (e.g. Table 3.2) and the manipulation commands (e.g. Fig. 3.8) contain object descriptions (**ObjDes**) for verbal refinement. If needed[3], this enables the image parsing to be updated during a manipulation operation. In Fig. 3.4 the distinction between commands 'refine' and 'correct' is as follows: the former should be given when the label assignment is good but the segment could be better; while, the later is to be given when the label is incorrect.

Consider that user give verbal command 'Refine the **ObjDes**', where

$$\textbf{ObjDes} = [\textbf{CA}][\textbf{SA}][\textbf{MA}]\textbf{Obj}[in\textbf{PA}]. \tag{3.8}$$

The system understands there should be an object named **Obj** in the position **PA**, and the correlation cues such as **MA**-**SA**, **MA**-**Obj** and **SA**-**Obj** should be encouraged. We achieve this by updating the correlation matrices given in (3.3). Thus, the altered object-attribute correlations are changed as $R'^{\mathcal{OA}} = \lambda_1 + \lambda_2 R^{\mathcal{OA}}$ and the modified attribute-attribute correlations are updated as $R'^{\mathcal{A}} = \lambda_3 + \lambda_4 R^{\mathcal{A}}$ where λ_i are tuning parameters.

[3]When we have perfect image parsing results for the image to be manipulated, we might verbally switch off the function that conducts this combination operation of image parsing and manipulation.

(a) source image (b) white $\mathbf{R}_c$ (c) center-middle $\mathbf{R}_s$

Figure 3.5: Response maps of $\mathbf{R}_c$ and $\mathbf{R}_s$ for attributes 'white' and 'center-middle' respectively.

Speech parsing: We use the freely available Microsoft speech SDK [149] to convert a spoken command into text. We use a simple speech grammar, with a small number of fixed commands. Since the structure of our verbal commands and the candidate keywords list are fixed, the grammar definition API of Microsoft speech SDK allows us to robustly capture user speech commands. For more sophisticated speech recognition and parsing, see [122].

colours $\mathbf{R}_c$ and spatial $\mathbf{R}_s$ attributes response map: Colourss are powerful attributes that can significantly improve performance of object classification [217] and detection [101]. To incorporate colour into our system, we create a colour response map, with the value at the ith pixel defined according to the distance between the colour of this pixel I_i and a user specified colour $\mathbb{I}$. We use $R_c(i) = 1 - \|I_i - \mathbb{I}\|$, where each of the RGB colour channels are in the range [0,1]. We also utilize the location information present in the command to localize objects. Similar to colour, the spatial response map value at the ith pixel is defined as $R_s(i) = \exp(-\frac{d^2}{2\delta^2})$, where d is the distance between the pixel location and the user indicated position. In the implementation, we use $\delta^2 = 0.04$ with pixel coordinates in both directions normalised to [0,1]. Fig. 3.5 illustrates an example of colour and position attributes generated according to a given verbal command. The spatial and colour response maps are combined into a final overall map $R(i) = R_s(i)R_c(i)$ that is used to update per pixel terms in (3.9). Since rough colour and position names are typically quite inaccurate, we average the initial response values within each region generated by the unsupervised segmentation method [75] for better robustness. These response maps are normalised to the same range as other object classes' per pixel terms for comparable influence to the learned object per pixel terms.

We use these response maps to update the corresponding object and attribute per pixel terms, $\psi_i^{\mathcal{O}}(x_i), \psi_{i,a}^{\mathcal{A}}(y_{i,a})$ in (3.2). Specifically, we set

$$\psi'^{\mathcal{O}}_i(x_i) = \psi_i^{\mathcal{O}}(.) - \lambda_5 R(i), \text{ if } x_i = \mathbb{O} \tag{3.9}$$

where $\psi_i^{\mathcal{O}}(x_i)$ is the per pixel term for objects and $\mathbb{O}$ is the user specified object. Attribute terms are updated in a similar manner and share the same λ_5 parameter. The $\lambda_{1,..,5}$ parameters are set via cross validation. After these per pixel terms are reset, the inference is re-computed to obtain the updated image parsing result.

Working set selection for efficient interaction: Our CRF is factorized for efficient inference over the full set of object and attribute labels. However, since the time it takes to perform inference is dependent on the number of labels that are considered, the interaction may take much longer if there are many labels. To overcome this problem, a smaller working set of labels can be employed during interaction, guaranteeing a smooth user experience. Moreover, as observed in [197], the actual number of object classes present in an image is usually much smaller than the total number of object-classes considered (around a maximum of 8 out of 397 in the SUN database [230]). We exploit this observation by deriving the working set as the set of labels in the result of our automatic parsing parse and then updating it as required during interaction, for instance if the user mentions a label currently not in the subset. In our implementation this strategy gives an average timing of around 0.2-0.3 seconds per interaction, independent of the total number of labels considered.

3.4 Evaluation

aNYU **Dataset (attributes augmented** NYU**):** We created a dataset for our evaluation since per-pixel joint object and attributes segmentation is an emerging problem and there are only a few existing benchmarks[4]. In order to train our model and perform quantitative evaluation, we augment the widely used NYU indoor V2 dataset [193], through additional manual labeling of semantic attributes. Fig. 3.6 illustrates an example of ground truth labeling of this dataset. We use the NYU images with ground truth object class labeling, and split the dataset into 724 training images and 725 testing images. The list of object classes and attributes we use can be found in Section 3.3.3. We only use the RGB images from the NYU dataset although it provides depth images. Notice that each pixels in the ground truth images are marked with an object class label and a set of attributes labels (on average, 64.7% of them are non empty sets).

Quantitative evaluation for automatic image parsing: We conduct quantitative evaluation on aNYU dataset. Our approach consists of automatic joint objects-attributes image

[4] As also noted by [206], although the CORE dataset [72] contains object and attributes labels, each CORE image only contains a single foreground object, without background annotations.

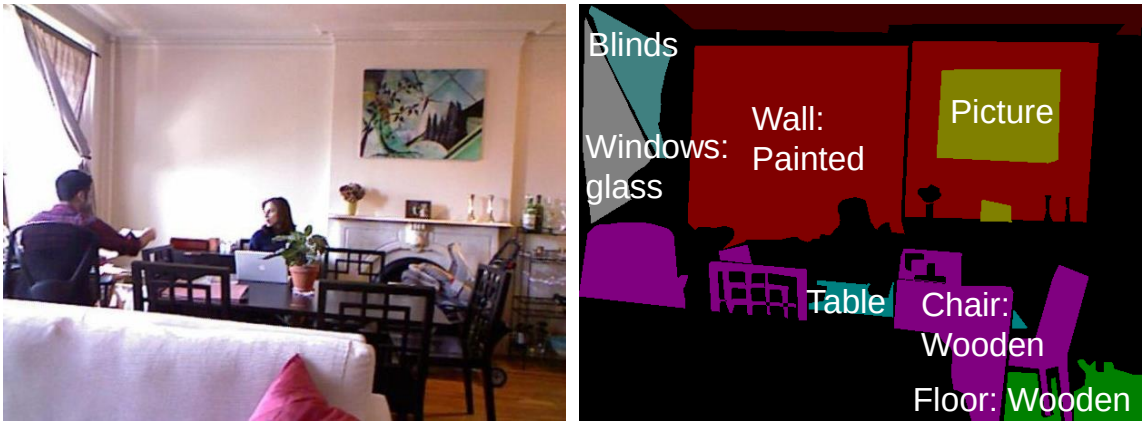
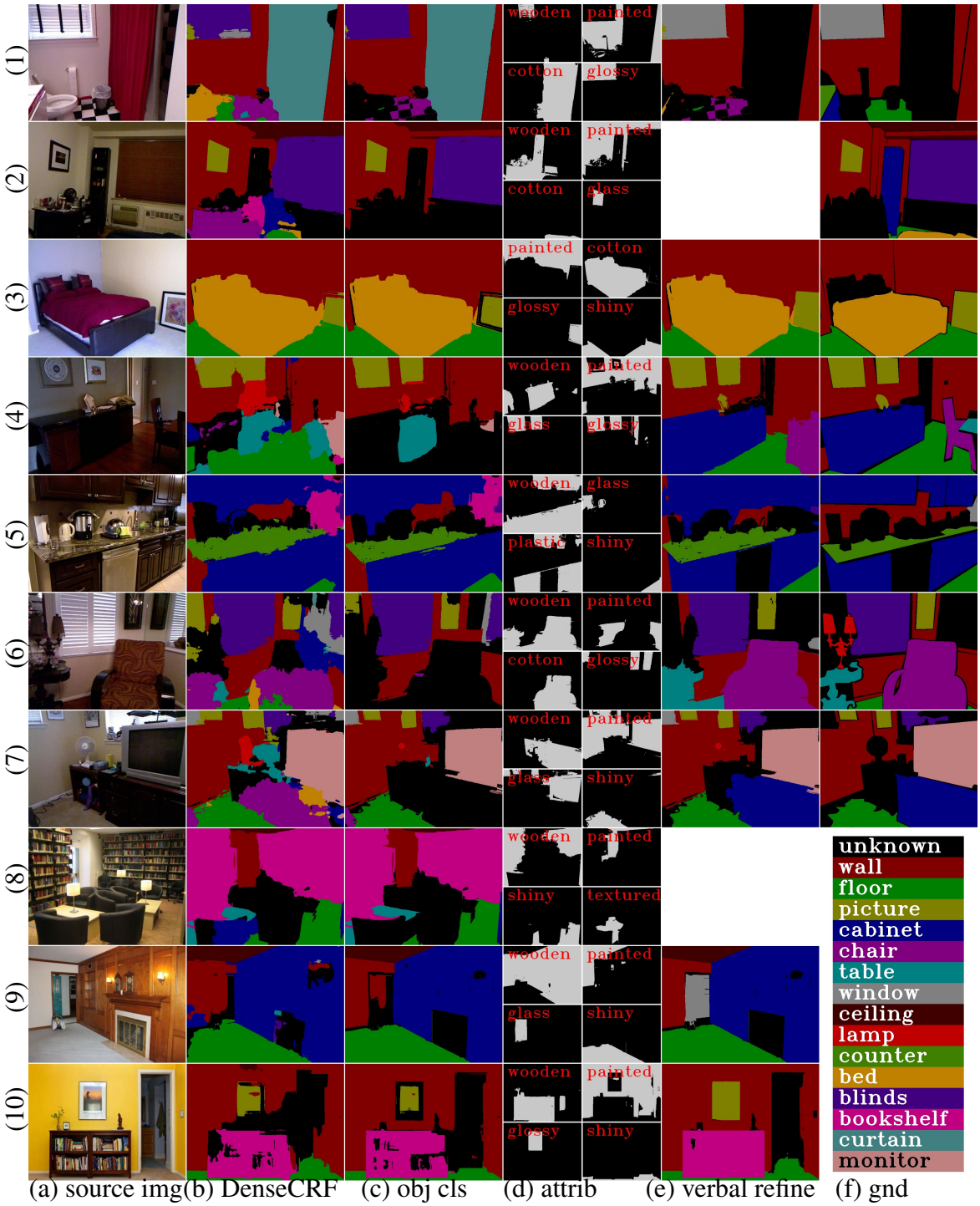


Figure 3.6: Example of ground truth labeling in aNYU dataset: original image (left) and object class and attributes labeling (right).

Table 3.2: Verbal commands used for parsing images in Fig. 3.7.

Image	Verbal commands
(1)	Correct the blinds to window. Correct the curtain to unknown.
(3)	Refine the glossy picture.
(4)	Refine the wooden cabinet in bottom-left. Refine the chair in bottom-right. Refine the floor in bottom-middle.
(5)	Refine the black plastic cabinet. Refine the white unknown in bottom-middle. Refine the cabinet in bottom-left.
(6)	Refine the cotton chair. Refine the glass unknown. Refine the black wooden table in bottom-left.
(7)	Refine the wooden cabinet in bottom-right.
(9)	Refine the glass window.
(10)	Refine the glossy picture. Refine the wooden bookshelf in bottom-middle. Refine the yellow painted wall in bottom middle. Refine the textured floor.

parsing and verbal guided image parsing. We compared our approach against two state-of-the-art CRF-based approaches including Associative Hierarchical CRF approach [116] and Dense CRF [110]. For fair comparison, we train the same TextonBoost classifiers for all the methods (a multi-class TextonBoost classifier for object class prediction and M independent binary TextonBoost classifiers, one for each attributes). Following [110], we adopt the average label accuracy (ALA) measure for algorithm performance which is the ratio between number of correctly labeled pixels and total number of pixels. As shown in Table 3.3, we have ALA score of 56.6% compared to 50.7% for the previous state-of-the-art results. During the experiments, we achieve best results when we set $T_a = 5$, as described in Algorithm 1.



(a) source img (b) DenseCRF (c) obj cls (d) attrib (e) verbal refine (f) gnd

Figure 3.7: Qualitative comparisons. Note that after verbal refinement, our algorithm provides results that correspond closely to human scene understanding. This is also reflected in the numerical results tabulated in Table 3.4. The last three images are from the Internet and lack ground truth. For the second and eight image, there are no attribute combinations which would improve the result, hence there is no verbal refinement. (See Table 3.2 for the used verbal commands.)

Table 3.3: Quantitative results on aNYU dataset. The H-CRF (Hierarchical conditional random field model) approach is implemented in a public available library: ALE), DenseCRF [110] represents the state-of-the-art CRF approach. Our-auto stands for our pixel-wise joint objects attributes image parsing approach. Our-inter means our verbally guided image parsing approach. All the experiments are carried out on a computer with Intel Xeon(E) 3.10GHz CPU and 12 GB RAM. Note that all methods in this table use the same features. Without the attributes terms, our CRF formulation will be reduced to exactly the same model as DenseCRF, showing that our JointCRF formulation benefits from the attributes components. Our-inter only considers the time used for updating the previous results given hints from user commands.

Methods	H-CRF	DenseCRF	Our-auto	Our-inter
Label accuracy	51.0%	50.7%	56.9%	- -
Inference time	13.2s	0.13s	0.54s	0.21s
Has attributes	NO	NO	YES	YES

Table 3.4: Evaluation for verbal guided image parsing. Here we show average statistics for interacting with a 50 images subset.

Methods	DenseCRF	Our-auto	Our-inter
Label accuracy	52.1%	56.2%	80.6%

Quantitative evaluation for verbal guided image parsing: We numerically evaluate our verbal guided interaction. We choose a subset of 50 images whose collective accuracy scores are reflective of the overall data set. After verbal refinement, our accuracy rises to 80.6% as compared to the $50 - 56\%$ of automated methods. From the results displayed in Fig. 3.7, one can see that these interactive improvements are not just numerical but also produce object segmentation that accord more to human intuition. In experiments, we achieve best speed-accuracy-trade-off results when we set $T_a = 5$, and $T_v = 3$, as described in Algorithm 1.

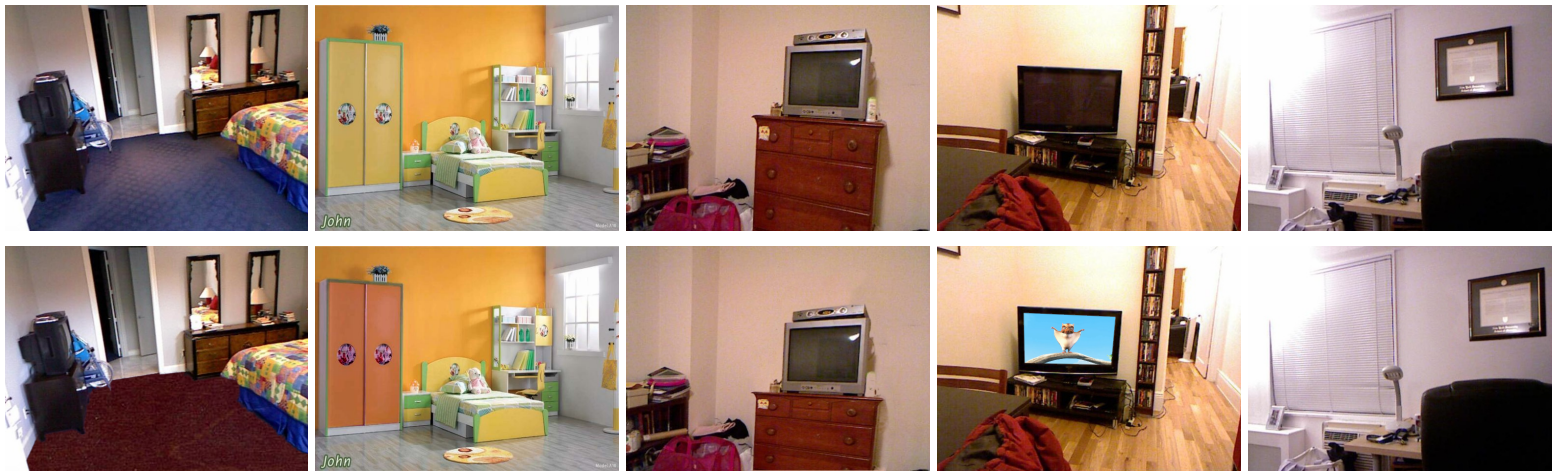
Note that the final 3 images of Fig. 3.7 are not part of the aNYU dataset but are Internet images without any ground truth annotations. These images demonstrate our algorithm's ability to generalize training data for application to images from a similar class (a system trained on indoor images will not work on outdoor scenes) taken under uncontrolled circumstances.

User study: Beyond large scale quantitative evaluation, we also test the plausibility of our new interaction modality by a user study. Our user study comprises of 38 participants, mostly computer science graduates. We investigate both the time efficiency and the user preference of the verbal interaction. Each user was given a one page instruction script and 1 minute demo video to show how to use verbal commands and mouse tools (line, brush, and

Table 3.5: Interactive time and accuracy comparison between different interaction modality: verbal, finger touch and both

Interaction modality	verbal	touch	verbal + touch
Average interaction time (s)	6.6	32.3	11.7
Average accuracy (%)	80.3	95.2	97.8
Average user preference (%)	15.8	10.5	73.7

fill tool as shown in Fig. 3.2) to interact with the system. The users were given 5 images and asked to improve the parsing results using different interaction modality: i) only verbal, ii) only finger touch, iii) both verbal and touch (in random order to reduce learning bias). Statistics about average interaction time, label accuracy, and user preference are shown in Table 3.5. In our experiments, participants use a small number of (mean and standard deviation: 1.6 ± 0.95) verbal commands to roughly improve the automatic parsing results and then touch interaction for further refinements. In the 'verbal+touch' modality, 73.7% users preferred verbal command before touch refinement. In desktop setting, although average preference of verbal interaction is not as good as touch interaction, it provides a viable alternative to touch interaction while the combination was generally preferred by most users. We believe that for new generation devices such as Google Glass and other wearable devices, our verbal interaction will be even more useful as it is not easy to perform traditional interactions on them.



(a) Re-Attributes (material) (b) Re-Attributes (colour) (c) Object deformation (d) Semantic animation (e) Re-Arrangement

Figure 3.8: Verbal guided image manipulation applications. The commands used are: (a) ‘Refine the white wall in bottom-left’ and ‘Change the floor to wooden’, (b) ‘Change the yellow wooden cabinet in center-left to brown’, (c) ‘Refine the glossy monitor’ and ‘Make the wooden cabinet lower’, (d) ‘Activate the black shiny monitor in center-middle’,

3.5 Manipulation Applications

To demonstrate our verbal guided system's applicability as a selection mechanism, we implement a hands-free image manipulation system. After scene parsing has properly segmented the desired object, we translate the verbs into pre-packaged sets of image manipulation commands. These commands include in-painting [198, 11] and alpha matting [127] needed for a seamless editing effect, as well as semantic rule-based considerations. The list of commands supported by our system is given in Fig. 3.4 and some sample results in Fig. 3.8. The detailed effects are given below. Although the hands-free image manipulation results are not entirely satisfactory, we believe that the initial results demonstrate the possibility offered by verbal scene parsing (see also video [5]).

Re-Attributes: Attributes, such as colour and surface properties have a large impact on object appearance. Changing these attributes is a common task and naturally lends itself to verbal control. Once the scene has been parsed, one can verbally specify the object to re-attribute. As the computer has pixel-wise knowledge of the region the user is referring too, it can apply the appropriate image processing operators to alter it. Among all the pixels with user specified object class label, we choose the 4-connected region with the biggest weight as the extent of the target object, with weights defined by the response map as shown in Fig. 3.5. Some examples are shown in Fig. 3.8. To change object colour, we add the difference between average colour of this object and the user specified target colour. For material changing, we simply tile the target texture (e.g. wood texture) within the object mask. Alternately, texture transfer methods [63] can be used. Note that in the current implementation, we ignore effects due to varying surface orientation.

Object Deformation and Re-Arrangement: Once an object has been accurately identified, our system supports move, size change and repeat commands that duplicate the object in a new region or changes its shape. Inpainting is automatically carried out to refill exposed regions. For robustness, we also define a simple, 'gravity' rule for the 'cabinet' and 'table' classes. This requires small objects above these object segments (except stuff such as wall and floor) to follow their motion. Note that without whole image scene parsing, this 'gravity' rule is difficult to implement as there is a concern that a background wall is defined as a small object. Examples of these move commands can be seen in Fig. 3.8c.

Semantic Animation: Real word objects often have their semantic functions. For example, a monitor could be used to display videos. Since we can estimate the object region and its semantic label, a natural manipulation would be animating these objects by a set of user or

[5] https://www.youtube.com/watch?v=-haAdPkzA3M

predefined animations. Our system supports an 'activate' command. By way of example consider Fig. 3.8, when the user says 'Activate the black shiny monitor in center-middle', our system automatically fits the monitor region with a rectangle shape, and shows a video in an detected inner rectangle of the full monitor boundary (typically related to screen area). This allows mimicking the real world function of the monitor class.

3.6 Discussion

This chapter presents a novel multi-label CRF formulation for efficient, image parsing into per-pixel object and attribute labels. The attribute labels act as verbal handles through which users can control the CRF, allowing verbal refinement of the image parsing. Despite the ambiguity of verbal descriptors, our system can deliver clearly image parsing results that correspond to human intuition. Such hands-free parsing of an image provides verbal methods to select objects of interest, which can then be used to aid image editing. Both the user study and the large scale quantitative evaluation verify the usefulness of our verbal parsing method. Our verbal interaction is especially suitable for new generation devices such as smart phones, Google Glass, consoles and living room devices. To encourage the research in this direction, we will release source code and benchmark datasets.

Limitations: Our approach has some limitations. Firstly, our reliance on attribute handles can fail if there is no combination of attributes that can be used to improve the image parsing. This can seen in the second and eighth image of Fig. 3.7 where we fail to provide any verbally refined result due to lack of appropriate attributes. Of the 78 images we tested (55 from dataset and 23 Internet images) only 10 (5 dataset and 5 Internet images) could not be further refined using attributes. This represents a 13% failure rate. Note that refinement failure does not imply overall failure and the automatic results may still be quite reasonable as seen in Fig. 3.7. Secondly, the ambiguity of language description prevents our algorithm from giving 100% accuracy.

Future work: Possible future directions might include extending our method to video analysis and inclusion of stronger physics based models as well as the use of more sophisticated techniques from machine learning. Interestingly our system can often segment objects that are not in our initial training set by relying solely on their attribute descriptions. In the future, we would like to better understand this effect and suitably select a canonical set of attributes to strengthen this functionality. It might also be interesting to explore efficient multi-class object detection algorithms to help working set selection, possibly supporting thousands of object classes [57, 47]. We have only scratched the surface of verbal guided

image parsing with many future possibilities, e.g., how to better combine touch and verbal commands, or how verbal refinement may change the learned models so that they perform better on further refinements.

3.7 Conclusion

In this chapter we developed a verbal guided image parsing system that allows users to refine the segmentation results by using verbal command. This application system is possible because the flexible of mean-field approximate inference algorithm.

Along with CRFs inference, there is a increasing interesting in neural networks such as convolutional neural networks and recurrent neural networks. Intuitively, the behavior mean-field approximate inference for CRFs is similar to recurrent neural networks. In next chapter, we show the same mean-field approximate inference can be reformulated as recurrent neural networks.

Chapter 4

Conditional Random Fields as Recurrent Neural Networks

Pixel-level labeling tasks, such as semantic segmentation, play a central role in image understanding. Previous two chapters are about generalising the semantic image segmentation for attributes and objects. However, the techniques there are based on the hand-craft features. Recent approaches have attempted to harness the capabilities of deep learning techniques for image recognition to tackle pixel-wise labeling tasks. One central issue in this methodology is the limited capacity of deep learning techniques to delineate visual objects. To solve this problem, in this chapter we introduce a new form of convolutional neural network that combines the strengths of Convolutional Neural Networks (CNNs) and Conditional Random Fields (CRFs)-based probabilistic graphical modeling. To this end, we formulate mean-field approximate inference for the Conditional Random Fields with Gaussian pairwise potentials as Recurrent Neural Networks. This network, called CRF-RNN, is then plugged in as a part of a CNN to obtain a deep network that has desirable properties of both CNNs and CRFs. Importantly, our system fully integrates CRF modeling with CNNs, making it possible to train the whole deep network end-to-end with the general back-propagation algorithm, avoiding offline post-processing methods for object delineation. We apply the proposed method to the problem of semantic image segmentation, obtaining top results on the challenging Pascal VOC 2012 segmentation benchmark.

4.1 Introduction

Low-level computer vision problems such as semantic image segmentation or depth estimation often involve assigning a label to each pixel in an image. While the feature representation used to classify individual pixels plays an important role in this task, it is similarly

important to consider factors such as image edges, appearance consistency and spatial consistency while assigning labels in order to obtain accurate and precise results.

Designing a strong feature representation is a key challenge in pixel-level labelling problems. Work on this topic includes: TextonBoost [192], TextonForest [191], and Random Forest-based classifiers [190]. Recently, supervised deep learning approaches such as large-scale deep Convolutional Neural Networks (CNNs) have been immensely successful in many high-level computer vision tasks such as image recognition [113] and object detection [79]. This motivates exploring the use of CNNs for pixel-level labelling problems. The key insight is to learn a strong feature representation end-to-end for the pixel-level labelling task instead of hand-crafting features with heuristic parameter tuning. In fact, a number of recent approaches including the particularly interesting works FCN [143] and DeepLab [34] have shown a significant accuracy boost by adapting state-of-the-art CNN based image classifiers to the semantic segmentation problem.

However, there are significant challenges in adapting CNNs designed for high level computer vision tasks such as object recognition to pixel-level labelling tasks. Firstly, traditional CNNs have convolutional filters with large receptive fields and hence produce coarse outputs when restructured to produce pixel-level labels [143]. Presence of max-pooling layers in CNNs further reduces the chance of getting a fine segmentation output [34]. This, for instance, can result in non-sharp boundaries and blob-like shapes in semantic segmentation tasks. Secondly, CNNs lack smoothness constraints that encourage label agreement between similar pixels, and spatial and appearance consistency of the labelling output. Lack of such smoothness constraints can result in poor object delineation and small spurious regions in the segmentation output [214, 213, 116, 152].

On a separate track to the progress of deep learning techniques, probabilistic graphical models have been developed as effective methods to enhance the accuracy of pixel-level labelling tasks. In particular, Markov Random Fields (MRFs) and its variant Conditional Random Fields (CRFs) have achieved widespread success in this area [116, 110] and have become one of the most successful graphical models used in computer vision. The key idea of CRF inference for semantic labelling is to formulate the label assignment problem as a probabilistic inference problem that incorporates assumptions such as the label agreement between similar pixels. CRF inference is able to refine weak and coarse pixel-level label predictions to produce sharp boundaries and fine-grained segmentations. Therefore, intuitively, CRFs can be used to overcome the drawbacks in utilizing CNNs for pixel-level labelling tasks.

One way to utilize CRFs to improve the semantic labelling results produced by a CNN is to apply CRF inference as a post-processing step disconnected from the training of the

CNN [34]. Arguably, this does not fully harness the strength of CRFs since it is not integrated with the deep network. In this setup, the deep network is unaware of the CRF during the training phase.

In this chapter, we propose an end-to-end deep learning solution for the pixel-level semantic image segmentation problem. Our formulation combines the strengths of both CNNs and CRF based graphical models in one unified framework. More specifically, we formulate mean-field approximate inference for the dense CRF with Gaussian pairwise potentials as a Recurrent Neural Network (RNN) which can refine coarse outputs from a traditional CNN in the forward pass, while passing error differentials back to the CNN during training. Importantly, with our formulation, the whole deep network, which comprises a traditional CNN and an RNN for CRF inference, can be trained end-to-end utilizing the usual back-propagation algorithm.

Arguably, when properly trained, the proposed network should outperform a system where CRF inference is applied as a post-processing method on independent pixel-level predictions produced by a pre-trained CNN. Our experimental evaluation confirms that this indeed is the case. We evaluate the performance of our network on the popular Pascal VOC 2012 benchmark, achieving a new state-of-the-art accuracy of 74.7%.

4.2 Related Work

In this section we review approaches that make use of deep learning and CNNs for low-level computer vision tasks, with a focus on semantic image segmentation. A wide variety of approaches have been proposed to tackle the semantic image segmentation task using deep learning. These approaches can be categorized into two main strategies.

The first strategy is based on utilizing separate mechanisms for feature extraction, and image segmentation exploiting the edges of the image [5, 151]. One representative instance of this scheme is the application of a CNN for the extraction of meaningful features, and using superpixels to account for the structural pattern of the image. Two representative examples are [69, 151], where the authors first obtained superpixels from the image and then used a feature extraction process on each of them. The main disadvantage of this strategy is that errors in the initial proposals (e.g: super-pixels) may lead to poor predictions, no matter how good the feature extraction process is. Pinheiro and Collobert [166] employed an RNN to model the spatial dependencies during scene parsing. In contrast to their approach, we show that a typical graphical model such as a CRF can be formulated as an RNN to form a part of a deep network, to perform end-to-end training combined with a CNN.

The second strategy is to directly learn a nonlinear model from the images to the label map. This, for example, was shown in [66], where the authors replaced the last fully connected layers of a CNN by convolutional layers to keep spatial information. An important contribution in this direction is Long *et al.* [143], where Long *et al.* used the concept of fully convolutional networks, and the notion that top layers obtain meaningful features for object recognition whereas low layers keep information about the structure of the image, such as edges. In their work, connections from early layers to later layers were used to combine these cues. Bell *et al.* [13] and Chen *et al.* [34, 157] used a CRF to refine segmentation results obtained from a CNN. Bell *et al.* focused on material recognition and segmentation, whereas Chen *et al.* reported very significant improvements on semantic image segmentation. In contrast to these works, which employed CRF inference as a standalone post-processing step disconnected from the CNN training, our approach is an end-to-end trainable network that jointly learns the parameters of the CNN and the CRF in one unified deep network.

Works that use neural networks to predict structured output are found in different domains. For example, Do *et al.* [59] proposed an approach to combine deep neural networks and Markov networks for sequence labeling tasks. Another domain which benefits from the combination of CNNs and structured loss is handwriting recognition. In natural language processing, Yao *et al.* [233] shows that the performance of an RNN-based words tagger can be significantly improved by incorporating elements of the CRF model. In [14], the authors combined a CNN with Hidden Markov Models for that purpose, whereas more recently Peng *et al.* [164] used a modified version of CRFs. Related to this line of works, in [96] a joint CNN and CRF model was used for text recognition on natural images. Tompson *et al.* [210] showed the use of joint training of a CNN and an MRF for human pose estimation, while Chen *et al.* [35] focused on the image classification problem with a similar approach. Another prominent work is [80], in which the authors express deformable part models, a kind of MRF, as a layer in a neural network. In our approach, we cast a different graphical model as a neural network layer.

A number of approaches have been proposed for automatic learning of graphical model parameters and joint training of classifiers and graphical models. Barbu *et al.* [10] proposed a joint training of a MRF/CRF model together with an inference algorithm in their Active Random Field approach. Domke [60] advocated back-propagation based parameter optimization in graphical models when approximate inference methods such as mean-field and belief propagation are used. This idea was utilized in [102], where a binary dense CRF was used for human pose estimation. Similarly, Ross *et al.* [177] and Stoyanov *et al.* [196] showed how back-propagation through belief propagation can be used to optimize model

parameters. Ross *et al.* [80] in particular proposes an approach based on learning messages. Many of these ideas can be traced back to [204], which proposes unrolling message passing algorithms as simpler operations that could be performed within a CNN. In a different setup, Krähenbühl and Koltun [111] demonstrated automatic parameter tuning of dense CRF when a modified mean-field algorithm is used for inference. An alternative inference approach for dense CRF, not based on mean-field, is proposed in [236].

In contrast to the works described above, our approach shows that it is possible to formulate dense CRF as an RNN so that one can form an end-to-end trainable system for semantic image segmentation which combines the strengths of deep learning and graphical modelling.

After our initial publication of the technical report of this work on arXiv.org, a number of independent works [184, 134] appeared on arXiv.org presenting similar joint training approaches for semantic image segmentation.

4.3 Conditional Random Fields

In this section we provide a brief overview of Conditional Random Fields (CRF) for pixel-wise labelling and introduce the notation used in the chapter. A CRF, used in the context of pixel-wise label prediction, models pixel labels as random variables that form a Markov Random Field (MRF) when conditioned upon a global observation. The global observation is usually taken to be the image.

Let X_i be the random variable associated to pixel i, which represents the label assigned to the pixel i and can take any value from a pre-defined set of labels $\mathcal{L} = \{l_1, l_2, \ldots, l_L\}$. Let $\mathbf{X}$ be the vector formed by the random variables $X_1, X_2, \ldots, X_N$, where N is the number of pixels in the image. Given a graph $G = (V, E)$ containing vertices and edges, where each vertice is associated with a random variable. A graph is corresponding to a global observation (image) $\mathbf{I}$. The pair $(\mathbf{I}, \mathbf{X})$ can be modelled as a CRF characterized by a Gibbs distribution of the form $P(\mathbf{X} = \mathbf{x}|\mathbf{I}) = \frac{1}{Z(\mathbf{I})} \exp(-E(\mathbf{x}|\mathbf{I}))$. Here $E(\mathbf{x})$ is called the energy of the configuration $\mathbf{x} \in \mathcal{L}^N$ and $Z(\mathbf{I})$ is the partition function [119]. From now on, we drop the conditioning on $\mathbf{I}$ in the notation for convenience.

In the fully connected pairwise CRF model of [110], the energy of a label assignment $\mathbf{x}$ is given by:

$$E(\mathbf{x}) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j), \tag{4.1}$$

where the unary energy components $\psi_u(x_i)$ measure the inverse likelihood (and therefore, the cost) of the pixel i taking the label x_i, and pairwise energy components $\psi_p(x_i, x_j)$ mea-

Algorithm 2 Mean-field in dense CRFs [110], broken down to common CNN operations.

$Q_i(l) \leftarrow \frac{1}{Z_i(\mathbf{U})} \exp\left(U_i(l)\right)$ for all i ▷ Initialization

while not converged **do**

$\tilde{Q}_i^{(m)}(l) \leftarrow \sum_{j \neq i} k_G^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l)$ for all m ▷ Message Passing

$\check{Q}_i(l) \leftarrow \sum_m w^{(m)} \tilde{Q}_i^{(m)}(l)$ ▷ Weighting Filter Outputs

$\hat{Q}_i(l) \leftarrow \sum_{l' \in \mathcal{L}} \mu(l, l') \check{Q}_i(l')$ ▷ Compatibility Transform

$\breve{Q}_i(l) \leftarrow U_i(l) - \hat{Q}_i(l)$ ▷ Adding Unary Potentials

$Q_i \leftarrow \frac{1}{Z_i(Q(\mathbf{X}))} \exp\left(\breve{Q}_i(l)\right)$ ▷ Softmax Normalisation

end while

sure the cost of assigning labels x_i, x_j to pixels i, j simultaneously. The unary and pairwise potentials depend on the location, this equation omit the location notation for the simplicity. In our model, unary energies are obtained from a CNN, which, roughly speaking, predicts labels for pixels without considering the smoothness and the consistency of the label assignments. The pairwise energies provide an image data-dependent smoothing term that encourages assigning similar labels to pixels with similar properties. As was done in [110], we model pairwise potentials as weighted Gaussians:

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{M} w^{(m)} k_G^{(m)}(\mathbf{f}_i, \mathbf{f}_j), \tag{4.2}$$

where each $k_G^{(m)}$ for $m = 1, \ldots, M$, is a Gaussian kernel applied on feature vectors, $w^{(m)}$ represent the weight parameters for different filtered results. The feature vector of pixel i, denoted by $\mathbf{f}_i$, is derived from image features such as spatial location and RGB values [110]. We use the same features as in[110]. The function $\mu(.,.)$, called the label compatibility function, captures the compatibility between different pairs of labels as the name implies.

Minimizing the above CRF energy $E(\mathbf{x})$ yields the most probable label assignment $\mathbf{x}$ for the given image. Since this exact minimization is intractable, a mean-field approximation to the CRF distribution is used for approximate maximum posterior marginal inference. It consists in approximating the CRF distribution $P(\mathbf{X})$ by a simpler distribution $Q(\mathbf{X})$, which can be written as the product of independent marginal distributions, i.e., $Q(\mathbf{X}) = \prod_i Q_i(X_i)$. The steps of the iterative algorithm for approximate mean-field inference and its reformulation as an RNN are discussed next.

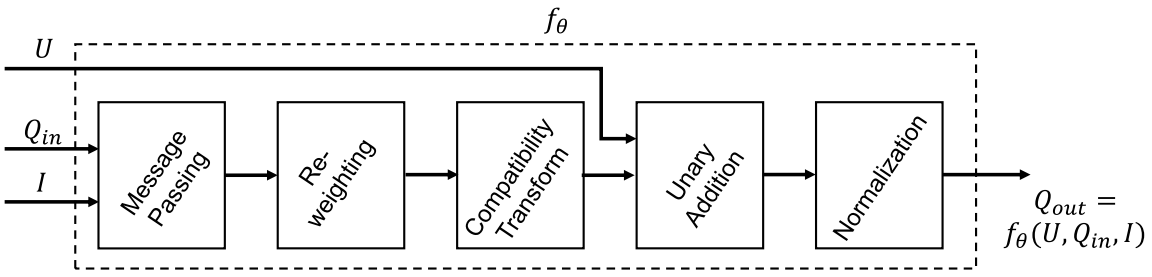


Figure 4.1: **A mean-field iteration as a CNN.** A single iteration of the mean-field algorithm can be modelled as a stack of common CNN layers.

4.4 A Mean-field Iteration as a Stack of CNN Layers

A key contribution of this chapter is to show that the mean-field CRF inference can be reformulated as a Recurrent Neural Network (RNN). To this end, we first consider individual steps of the mean-field algorithm summarized in Algorithm 2 [110], and describe them as CNN layers. Our contribution is based on the observation that filter-based approximate mean-field inference approach for dense CRFs relies on applying Gaussian spatial and bilateral filters on the mean-field approximates in each iteration. Unlike the standard convolutional layer in a CNN, in which filters are fixed after the training stage, we use edge-preserving Gaussian filters [209, 159], coefficients of which depend on the original spatial and appearance information of the image. These filters have the additional advantages of requiring a smaller set of parameters, despite the filter size being potentially as big as the image.

While reformulating the steps of the inference algorithm as CNN layers, it is essential to be able to calculate error differentials in each layer w.r.t. its inputs in order to be able to back-propagate the error differentials to previous layers during training. We also discuss how to calculate error differentials with respect to the parameters in each layer, enabling their optimization through the back-propagation algorithm. Therefore, in our formulation, CRF parameters such as the weights of the Gaussian kernels and the label compatibility function can also be optimized automatically during the training of the full network.

Once the individual steps of the algorithm are broken down as CNN layers, the full algorithm can then be formulated as an RNN. We explain this in Section 4.5 after discussing the steps of Algorithm 2 in detail below. In Algorithm 2 and the remainder of this chapter, we use $U_i(l)$ to denote the negative of the unary energy introduced in the previous section, i.e., $U_i(l) = -\psi_u(X_i = l)$. In the conventional CRF setting, this input $U_i(l)$ to the mean-field algorithm is obtained from an independent classifier.

4.4.1 Initialization

In the initialization step of the algorithm, the operation $Q_i(l) \leftarrow \frac{1}{Z_i}\exp\left(U_i(l)\right)$, where $Z_i = \sum_l \exp(U_i(l))$, is performed. Note that this is equivalent to applying a softmax function over the unary potentials U across all the labels at each pixel. The softmax function has been extensively used in CNN architectures before and is therefore well known in the deep learning community. This operation does not include any parameters and the error differentials received at the output of the step during back-propagation could be passed down to the unary potential inputs after performing usual backward pass calculations of the softmax transformation.

4.4.2 Message Passing

In the dense CRF formulation, message passing is implemented by applying M Gaussian filters to the Q values. Gaussian filter coefficients are derived based on image features such as the pixel locations and RGB values, which reflect how strongly a pixel is related to other pixels. Since the CRF is potentially fully connected, each filter's receptive field spans the whole image, making it infeasible to use a brute-force implementation of the filters. Fortunately, several approximation techniques exist to make computation of high dimensional Gaussian filtering significantly faster. Following [110], we use the Permutohedral lattice implementation [3], which can compute the filter response in $O(N)$ time, where N is the number of pixels of the image [3].

During back-propagation, error derivatives w.r.t. the filter inputs are calculated by sending the error derivatives w.r.t. the filter outputs through the same M Gaussian filters in reverse direction. In terms of permutohedral lattice operations, this can be accomplished by only reversing the order of the separable filters in the blur stage, while building the permutohedral lattice, splatting, and slicing in the same way as in the forward pass. Therefore, back-propagation through this filtering stage can also be performed in $O(N)$ time. Following [110], we use two Gaussian kernels, a spatial kernel and a bilateral kernel. In this work, for simplicity, we keep the bandwidth values of the filters fixed. It is also possible to use multiple spatial and bilateral kernels with different bandwidth values and learn their optimal linear combination.

4.4.3 Weighting Filter Outputs

The next step of the mean-field iteration is taking a weighted sum of the M filter outputs from the previous step, for each class label l. When each class label is considered individually, this can be viewed as usual convolution with a 1×1 filter with M input channels,

and one output channel. Since both inputs and the outputs to this step are known during back-propagation, the error derivative w.r.t. the filter weights can be computed, making it possible to automatically learn the filter weights (relative contributions from each Gaussian filter output from the previous stage). Error derivatives w.r.t. the inputs can also be computed in the usual manner to pass the error derivatives down to the previous stage. To obtain a higher number of tunable parameters, in contrast to [110], we use independent kernel weights for each class label. The intuition is that the relative importance of the spatial kernel vs the bilateral kernel depends on the visual class. For example, bilateral kernels may have on the one hand a high importance in bicycle detection, because similarity of colours is determinant; on the other hand they may have low importance for TV detection, given that whatever is inside the TV screen may have many different colours.

4.4.4 Compatibility Transform

In the compatibility transform step, outputs from the previous step (denoted by $\check{Q}$ in Algorithm 2) are shared between the labels to a varied extent, depending on the compatibility between these labels. Compatibility between the two labels l and l' is parameterized by the label compatibility function $\mu(l, l')$. The Potts model, given by $\mu(l, l') = [l \neq l']$, where $[.]$ is the Iverson bracket, assigns a fixed penalty if different labels are assigned to pixels with similar properties. A limitation of this model is that it assigns the same penalty for all different pairs of labels. Intuitively, better results can be obtained by taking the compatibility between different label pairs into account and penalizing the assignments accordingly. For example, assigning labels "person" and "bicycle" to nearby pixels should have a lesser penalty than assigning labels "sky" and "bicycle". Therefore, learning the function μ from data is preferred to fixing it in advance with the Potts model. We also relax our compatibility transform model by assuming that $\mu(l, l') \neq \mu(l', l)$ in general.

The compatibility transform step can be viewed as another convolution layer where the spatial receptive field of the filter is 1×1, and the numbers of input and output channels are both L. Learning the weights of this filter is equivalent to learning the label compatibility function μ. Transferring error differentials from the output of this step to the input can be done since this step is a usual convolution operation.

4.4.5 Adding Unary Potentials

In this step, the output from the compatibility transform stage is subtracted element-wise from the unary inputs U. While no parameters are involved in this step, transferring error

differentials can be done trivially by copying the differentials at the output of this step to both inputs with the appropriate sign.

4.4.6 Normalisation

Finally, the normalisation step of the iteration can be considered as another softmax operation with no parameters. Differentials at the output of this step can be passed on to the input using the softmax operation's backward pass.

4.5 The End-to-end Trainable Network

We now describe our end-to-end deep learning system for semantic image segmentation. To pave the way for this, we first explain how repeated mean-field iterations can be organized as an RNN.

4.5.1 CRF as RNN

In the previous section, it was shown that one iteration of the mean-field algorithm can be formulated as a stack of common CNN layers (see Fig. 4.1). We use the function $f_{\boldsymbol{\theta}}$ to denote the transformation done by one mean-field iteration: given an image I, pixel-wise unary potential values U and an estimation of marginal probabilities Q_{in} from the previous iteration, the next estimation of marginal distributions after one mean-field iteration is given by $f_{\boldsymbol{\theta}}(U, Q_{\text{in}}, I)$. The vector $\boldsymbol{\theta} = \{w^{(m)}, \mu(l, l')\}, m \in \{1, ..., M\}, l, l' \in \{l_1, ..., l_L\}$ represents the CRF parameters described in Section 4.4.

Multiple mean-field iterations can be implemented by repeating the above stack of layers in such a way that each iteration takes Q value estimates from the previous iteration and the unary values in their original form. This is equivalent to treating the iterative mean-field inference as a Recurrent Neural Network (RNN) as shown in Fig. 4.2. Using the notation in the figure, the behaviour of the network and the gating functions are given by the following equations where T is the number of mean-field iterations:

$$H_1(t) = \begin{cases} \text{softmax}(U), & t = 0 \\ H_2(t-1), & 0 < t \leq T, \end{cases} \tag{4.3}$$

$$H_2(t) = f_{\boldsymbol{\theta}}(U, H_1(t), I), \quad 0 \leq t \leq T, \tag{4.4}$$

$$Y(t) = \begin{cases} 0, & 0 \leq t < T \\ H_2(t), & t = T. \end{cases} \tag{4.5}$$

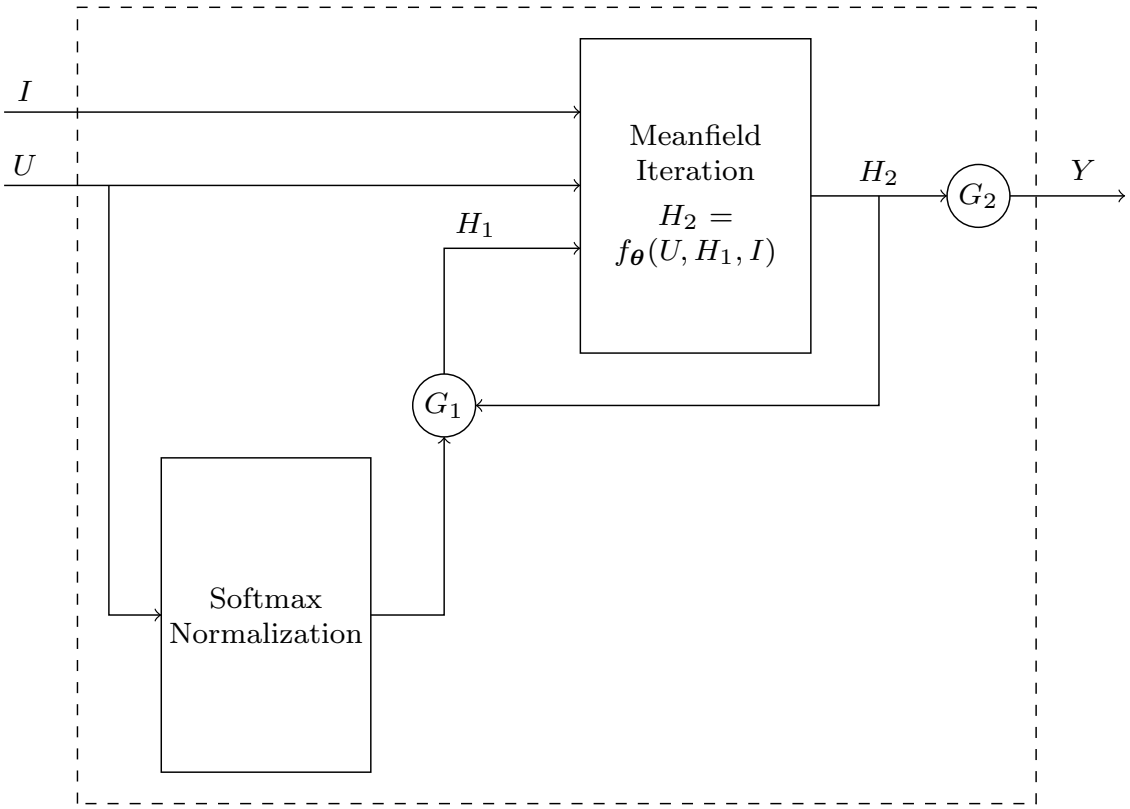


Figure 4.2: **The CRF-RNN Network.** We formulate the iterative mean-field algorithm as a Recurrent Neural Network (RNN). Gating functions G_1 and G_2 are fixed as described in the text.

We name this RNN structure CRF-RNN. Parameters of the CRF-RNN are the same as the mean-field parameters described in Section 4.4 and denoted by $\boldsymbol{\theta}$ here. Since the calculation of error differentials w.r.t. these parameters in a single iteration was described in Section 4.4, they can be learnt in the RNN setting using the standard back-propagation through time algorithm [182, 153]. It was shown in [110] that the mean-field iterative algorithm for dense CRF converges in less than 10 iterations. Furthermore, in practice, after about 5 iterations, increasing the number of iterations usually does not significantly improve results [110]. Therefore, it does not suffer from the vanishing and exploding gradient problem inherent to deep RNNs [15, 160]. This allows us to use a plain RNN architecture instead of more sophisticated architectures such as LSTMs in our network.

4.5.2 Completing the Picture

Our approach comprises a fully convolutional network stage, which predicts pixel-level labels without considering structure, followed by a CRF-RNN stage, which performs CRF-

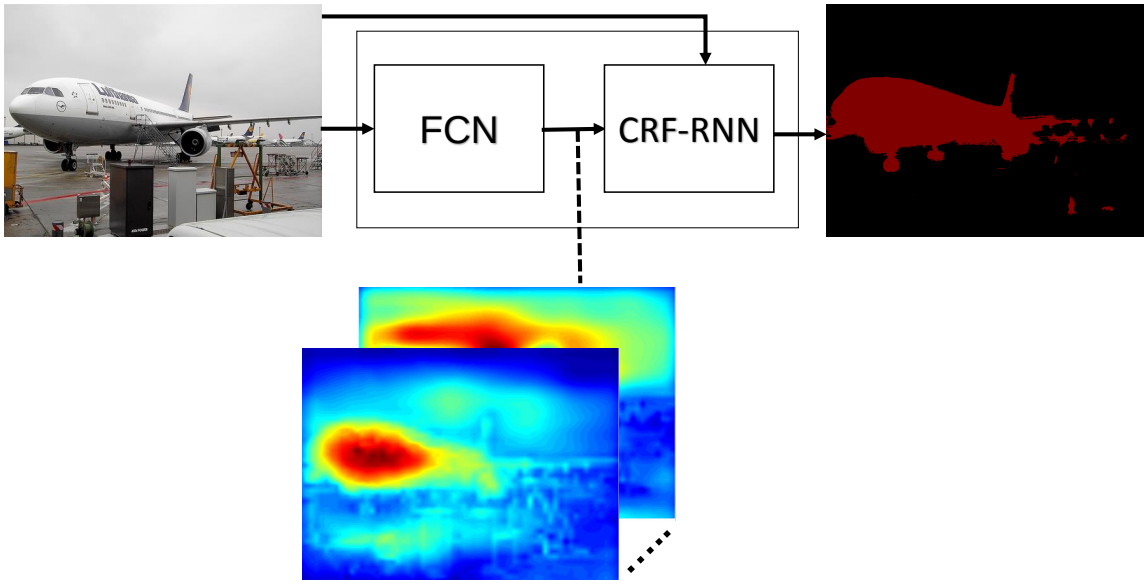


Figure 4.3: **The End-to-end Trainable Network.** Schematic visualization of our full network which consists of a CNN and the CNN-CRF network. Best viewed in colour.

based probabilistic graphical modelling for structured prediction. The complete system, therefore, unifies strengths of both CNNs and CRFs and is trainable end-to-end using the back-propagation algorithm [125] and the Stochastic Gradient Descent (SGD) procedure. During training, a whole image (or many of them) can be used as the mini-batch and the error at each pixel output of the network can be computed using an appropriate loss function such as the softmax loss with respect to the ground truth segmentation of the image. We used the FCN-8s architecture of [143] as the first part of our network, which provides unary potentials to the CRF. This network is based on the VGG-16 network [195] but has been restructured to perform pixel-wise prediction instead of image classification.

In the forward pass through the network, once the computation enters the CRF-RNN after passing through the CNN stage, it takes T iterations for the data to leave the loop created by the RNN. Neither the CNN that provides unary values nor the layers after the CRF-RNN (i.e., the loss layers) need to perform any computations during this time since the refinement happens only inside the RNN's loop. Once the output Y leaves the loop, next stages of the deep network after the CRF-RNN can continue the forward pass. In our setup, a softmax loss layer directly follows the CRF-RNN and terminates the network.

During the backward pass, once the error differentials reach the CRF-RNN's output Y, they similarly spend T iterations within the loop before reaching the RNN input U in order to propagate to the CNN which provides the unary input. In each iteration inside the loop, error differentials are computed inside each component of the mean-field iteration as

described in Section 4.4. We note that unnecessarily increasing the number of mean-field iterations T could potentially result in the vanishing and exploding gradient problems in the CRF-RNN. We, however, did not experience this problem during our experiments.

4.6 Implementation Details

In the present section we describe the implementation details of the proposed network, as well as its training process. The high-level architecture of our system, which was implemented using the popular Caffe [98] deep learning library, is shown in Fig. 4.3. The full source code and the trained models of our approach are publicly available [1].

We initialized the first part of the network using the publicly available weights of the FCN-8s network [143]. The compatibility transform parameters of the CRF-RNN were initialized using the Potts model, and kernel width and weight parameters were obtained from a cross-validation process. We found that such initialization results in faster convergence of training. During the training phase, parameters of the whole network were optimized end-to-end using the back-propagation algorithm. In particular we used full image training described in [143], with learning rate fixed at 10^{-13} and momentum set to 0.99. These extreme values of the parameters were used since we employed only one image per batch to avoid reaching memory limits of the GPU.

In all our experiments, during training, we set the number of mean-field iterations T in the CRF-RNN to 5 to avoid vanishing/exploding gradient problems and to reduce the training time. During the test time, iteration count was increased to 10. The effect of this parameter value on the accuracy is discussed in section 4.7.1.

Loss function During the training of the models that achieved the best results reported in this chapter, we used the standard softmax loss function, that is, the log-likelihood error function described in [111]. The standard metric used in the Pascal VOC challenge is the average intersection over union (IU), which we also use here to report the results. In our experiments we found that high values of IU on the validation set were associated to low values of the averaged softmax loss, to a large extent. We also tried the robust log-likelihood in [111] as a loss function for CRF-RNN training. However, this did not result in increased accuracy nor faster convergence.

Normalisation techniques As described in Section 4.4, we use the exponential function followed by pixel-wise normalisation across channels in several stages of the CRF-RNN. Since this operation has a tendency to result in small gradients with respect to the input when the input value is large, we conducted several experiments where we replaced

[1] https://github.com/torrvision/crfasrnn/.

this by a rectifier linear unit (ReLU) operation followed by a normalisation across the channels. Our hypothesis was that this approach might approximate the original operation adequately while speeding up the training due to improved gradients. Furthermore, ReLU would induce sparsity on the probability of labels assigned to pixels, implicitly pruning low likelihood configurations, which could have a positive effect. However, this approach did not lead to better results, obtaining 1% IU lower than the original setting performance.

4.7 Experiments

We present experimental results with the proposed CRF-RNN framework. We use these datasets: the Pascal VOC 2012 dataset, and the Pascal Context dataset. We use the Pascal VOC 2012 dataset as it has become the golden standard to comprehensively evaluate any new semantic segmentation approach in comparison to existing methods. We also use the Pascal Context dataset to assess how well our approach performs on a dataset with different characteristics.

Pascal VOC Datasets

In order to evaluate our approach with existing methods under the same circumstances, we conducted two main experiments with the Pascal VOC 2012 dataset, followed by a qualitative experiment.

In the first experiment, following [143, 151, 157], we used a training set consisted of VOC 2012 training data (1464 images), and training and validation data of [88], which amounts to a total of 11,685 images. After removing the overlapping images between VOC 2012 validation data and this training dataset, we were left with 346 images from the original VOC 2012 validation set to validate our models on. We call this set the reduced validation set in the sequel. Annotations of the VOC 2012 test set, which consists of 1456 images, are not publicly available and hence the final results on the test set were obtained by submitting the results to the Pascal VOC challenge evaluation server [67]. Regardless of the smaller number of images, we found that the relative improvements of the accuracy on our validation set were in good agreement with the test set.

As a first step we directly compared the potential advantage of learning the model end-to-end with respect to alternative learning strategies. These are plain FCN-8s without applying CRF, and with CRF as a postprocessing method disconnected from the training of FCN, which is comparable to the approach described in [34] and [157]. The results are reported in Table 4.1 and show a clear advantage of the end-to-end strategy over the offline

Figure 4.4: **Qualitative results on the validation set of Pascal VOC 2012.** FCN [143] is a CNN-based model that does not employ CRF. Deeplab [34] is a two-stage approach, where the CNN is trained first, and then CRF is applied on top of the CNN output. Our approach is an end-to-end trained system that integrates both CNN and CRF-RNN in one deep network. Best viewed in colour.

application of CRF as a post-processing method. This can be attributed to the fact that during the SGD training of the CRF-RNN, the CNN component and the CRF component learn

how to co-operate with each other to produce the optimum output of the whole network.

We then proceeded to compare our approach with all state-of-the-art methods that used training data from the standard VOC 2012 training and validation sets, and from the dataset published with [87]. The results are shown in Table 4.2, above the bar, and we can see that our approach outperforms all competitors.

In the second experiment, in addition to the above training set, we used data from the Microsoft COCO dataset [135] as was done in [157] and [53]. We selected images from MS COCO 2014 training set where the ground truth segmentation has at least 200 pixels marked with classes labels present in the VOC 2012 dataset. With this selection, we ended up using 66,099 images from the COCO dataset and therefore a total of 66,099 + 11,685 = 77,784 training images were used in the second experiment. The same reduced validation set was used in this second experiment as well. In this case, we first fine-tuned the plain FCN-32s network (without the CRF-RNN part) on COCO data, then we built an FCN-8s network with the learnt weights and finally train the CRF-RNN network end-to-end using VOC 2012 training data only. Since the MS COCO ground truth segmentation data contains somewhat coarse segmentation masks where objects are not delineated properly, we found that fine-tuning our model with COCO did not yield significant improvements. This can be understood because the primary advantage of our model comes from delineating the objects and improving fine segmentation boundaries. The VOC 2012 training dataset therefore helps our model learn this task effectively. The results of this experiment are shown in Table 4.2, below the bar, and we see that our approach sets a new state-of-the-art on the VOC 2012 dataset.

Note that in both setups, our approach outperforms competing methods due to the end-to-end training of the CNN and CRF in the unified CRF-RNN framework. We also evaluated our models on the VOC 2010, and VOC 2011 test set (see Table 4.2). In all cases our method achieves the state-of-the-art performance.

In order to have a qualitative evidence about how CRF-RNN learns, we visualize the compatibility function learned after the training stage of the CRF-RNN as a matrix representation in Fig. 4.5. Element (i, j) of this matrix corresponds to $\mu(i, j)$ defined earlier: a high value at (i, j) implies high penalty for assigning label i to a pixel when a similar pixel (spatially or appearance wise) is assigned label j. For example we can appreciate that the learned compatibility matrix assigns a low penalty to pairs of labels that tend to appear together, such as [*Motorbike*, *Person*], and [*Dining table*, *Chair*].

Method	Without COCO	With COCO
Plain FCN-8s	61.3	68.3
FCN-8s and CRF disconnected	63.7	69.5
End-to-end training of CRF-RNN	69.6	72.9

Table 4.1: Mean IU accuracy of our approach, CRF-RNN, compared with similar methods, evaluated on the reduced VOC 2012 validation set.

Method	VOC 2010 test	VOC 2011 test	VOC 2012 test
BerkeleyRC [6]	n/a	39.1	n/a
O2PCPMC [30]	49.6	48.8	47.8
Divmbest [163]	n/a	n/a	48.1
NUS-UDS [61]	n/a	n/a	50.0
SDS [88]	n/a	n/a	51.6
MSRA-CFM [54]	n/a	n/a	61.8
FCN-8s [143]	n/a	62.7	62.2
Hypercolumn [89]	n/a	n/a	62.6
Zoomout [151]	64.4	64.1	64.4
Context-Deep-CNN-CRF [134]	n/a	n/a	70.7
DeepLab-MSc [34]	n/a	n/a	71.6
Our method w/o COCO	73.6	72.4	72.0
BoxSup [53]	n/a	n/a	71.0
DeepLab [34, 157]	n/a	n/a	72.7
Our method with COCO	**75.7**	**75.0**	**74.7**

Table 4.2: Mean IU accuracy of our approach, CRF-RNN, compared to the other approaches on the Pascal VOC 2010-2012 test datasets. Methods from the first group do not use MS COCO data for training. The methods from the second group use both COCO and VOC datasets for training.

Pascal Context Dataset

We conducted an experiment on the Pascal Context dataset [152], which differs from the previous one in the larger number of classes considered, 59. We used the provided partitions

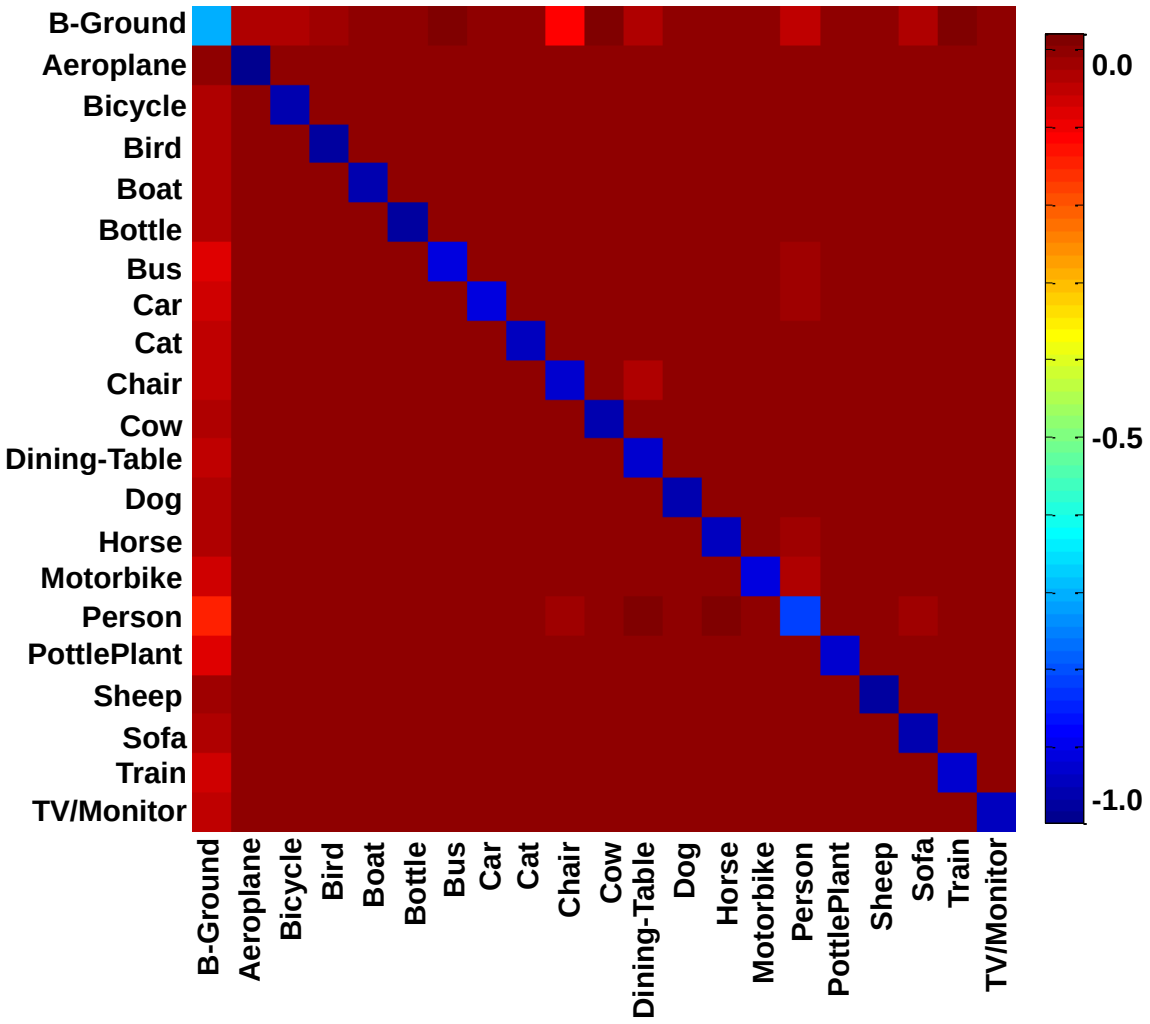


Figure 4.5: **Visualization of the learnt label compatibility matrix.** In the standard Potts model, diagonal entries are equal to -1, while off-diagonal entries are zero. These values have changed after the end-to-end training of our network. Best viewed in colour.

of training and validation sets, and the obtained results are reported in Table 4.3.

4.7.1 Effect of Design Choices

We performed a number of additional experiments on the Pascal VOC 2012 validation set described above to study the effect of some design choices we made.

We first studied the performance gains attained by our modifications to the CRF over the CRF approach proposed by [110]. We found that using different filter weights for different classes improved the performance by 1.8 percentage points, and that introducing the asymmetric compatibility transform further boosted the performance by 0.9 percentage points.

Regarding the RNN parameter iteration count T, incrementing it to $T = 10$ during the test time, from $T = 5$ during the train time, produced an accuracy improvement of 0.2 percentage points. Setting $T = 10$ also during training reduced the accuracy by 0.7 percentage points. We believe that this might be due to a vanishing gradient effect caused by using too many iterations. In practice that leads to the first part of the network (the one producing unary potentials) receiving a very weak error gradient signal during training, thus hampering its learning capacity.

End-to-end training after the initialization of CRF parameters improved performance by 3.4 percentage points. We also conducted an experiment where we froze the FCN-8s part and fine-tuned only the RNN part (i.e., CRF parameters). It improved the performance over initialization by only 1 percentage point. We therefore conclude that end-to-end training significantly contributed to boost the accuracy of the system.

Treating each iteration of mean-field inference as an independent step with its own parameters, and training end-to-end with 5 such iterations yielded a final mean IU score of only 70.9, supporting the hypothesis that the recurrent structure of our approach is important for its success.

4.8 Conclusion

We presented CRF-RNN, an interpretation of dense CRFs as Recurrent Neural Networks. Our formulation fully integrates CRF-based probabilistic graphical modelling with emerging deep learning techniques. In particular, the proposed CRF-RNN can be plugged in as a part of a traditional deep neural network: It is capable of passing on error differentials from its outputs to inputs during back-propagation based training of the deep network while learning CRF parameters. We demonstrate the use of this approach by utilizing it for the semantic segmentation task: we form an end-to-end trainable deep network by combining a fully convolutional neural network with the CRF-RNN. Our system achieves a new state-of-the-art on the popular Pascal VOC segmentation benchmark. This improvement can be attributed to the uniting of the strengths of CNNs and CRFs in a single deep network.

Method	O_2P [30]	CFM [54]	FCN-8s [143]	CRF-RNN
Mean IU	18.1	34.4	37.78	**39.28**

Table 4.3: Mean IU accuracy of our approach, CRF-RNN, evaluated on the Pascal Context validation set.

In the future, we plan to investigate the advantages/disadvantages of restricting the capabilities of the RNN part of our network to mean-field inference of dense CRF. A sensible baseline to the work presented here would be to use more standard RNNs (*e.g.* LSTMs) that learn to iteratively improve the input unary potentials to make them closer to the ground-truth.

Mean-field approximate inference and fully-connected conditional random fields are interesting because its efficiency. In literature, semantic image segmentation is addressed with GraphCut algorithm. In next chapter, we found the two different algorithms are equivalent.

Methods trained with COCO	Mean IU									
Our method	**74.7**	**90.4**	**55.3**	**88.7**	**68.4**	69.8	88.3	82.4	85.1	32.6
DeepLab[34, 157]	72.7	89.1	38.3	88.1	63.3	69.7	87.1	83.1	85.0	29.3
BoxSup[53]	71.0	86.4	35.5	79.7	65.2	65.2	84.3	78.5	83.7	30.5
Methods trained w/o COCO										
Our method trained w/o COCO	72.0	87.5	39.0	79.7	64.2	68.3	87.6	80.8	84.4	30.4
DeepLab-MSc-CRF-LargeFOV[34]	71.6	84.4	54.5	81.5	63.6	65.9	85.1	79.1	83.4	30.7
Context_Deep_CNN_CRF[134]	70.7	87.5	37.7	75.8	57.4	72.3	88.4	82.6	80.0	33.4
Zoomout[151]	64.4	81.9	35.1	78.2	57.4	56.5	80.5	74.0	79.8	22.4
Hypercolumn[89]	62.6	68.7	33.5	69.8	51.3	70.2	81.1	71.9	74.9	23.9
FCN-8s[143]	62.2	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4
MSRA_CFM[54]	61.8	75.7	26.7	69.5	48.8	65.6	81.0	69.2	73.3	30.0
SDS[88]	51.6	63.3	25.7	63.0	39.8	59.2	70.9	61.4	54.9	16.8
NUS_UDS [61]	50.0	67.0	24.5	47.2	45.0	47.9	65.3	60.6	58.5	15.5
TTIC-divmbest-rerank[163]	48.1	62.7	25.6	46.9	43.0	54.8	58.4	58.6	55.6	14.6
BONN_O2PCPMC_FGT_SEGM [30]	47.8	64.0	27.3	54.1	39.2	48.7	56.6	57.7	52.5	14.2

Methods trained with COCO											
Our method	**78.5**	**64.4**	79.6	**81.9**	**86.4**	81.8	58.6	82.4	53.5	77.4	**70.1**
DeepLab[34, 157]	76.5	56.5	79.8	77.9	85.8	82.4	57.4	84.3	54.9	80.5	64.1
BoxSup[53]	76.2	62.6	79.3	76.1	82.1	81.3	57.0	78.2	55.0	72.5	68.1
Methods trained w/o COCO											
Our method trained w/o COCO	78.2	60.4	80.5	77.8	83.1	80.6	59.5	82.8	47.8	78.3	67.1
DeepLab-MSc-CRF-LargeFOV [34]	74.1	59.8	79.0	76.1	83.2	80.8	59.7	82.2	50.4	73.1	63.7
Context_Deep_CNN_CRF[134]	71.5	55.0	79.3	78.4	81.3	82.7	56.1	79.8	48.6	77.1	66.3
TTI_zoomout_16[151]	69.6	53.7	74.0	76.0	76.6	68.8	44.3	70.2	40.2	68.9	55.3
Hypercolumn[89]	60.6	46.9	72.1	68.3	74.5	72.9	52.6	64.4	45.4	64.9	57.4
FCN-8s[143]	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1
MSRA_CFM[54]	68.7	51.5	69.1	68.1	71.7	67.5	50.4	66.5	44.4	58.9	53.5
SDS[88]	45.0	48.2	50.5	51.0	57.7	63.3	31.8	58.7	31.2	55.7	48.5
NUS_UDS[61]	50.8	37.4	45.8	59.9	62.0	52.7	40.8	48.2	36.8	53.1	45.6
TTIC-divmbest-rerank[163]	47.5	31.2	44.7	51.0	60.9	53.5	36.6	50.9	30.1	50.2	46.8
BONN_O2PCPMC_FGT_SEGM[30]	54.8	29.6	42.2	58.0	54.8	50.2	36.6	58.6	31.6	48.4	38.6

Table 4.4: Intersection over Union (IU) accuracy of our approach, CRF-RNN, compared to the other state-of-the-art approaches on the Pascal VOC 2012 test set. Scores for other methods were taken the results published by the original authors. The symbols are from Chatfield *et al.* [33].

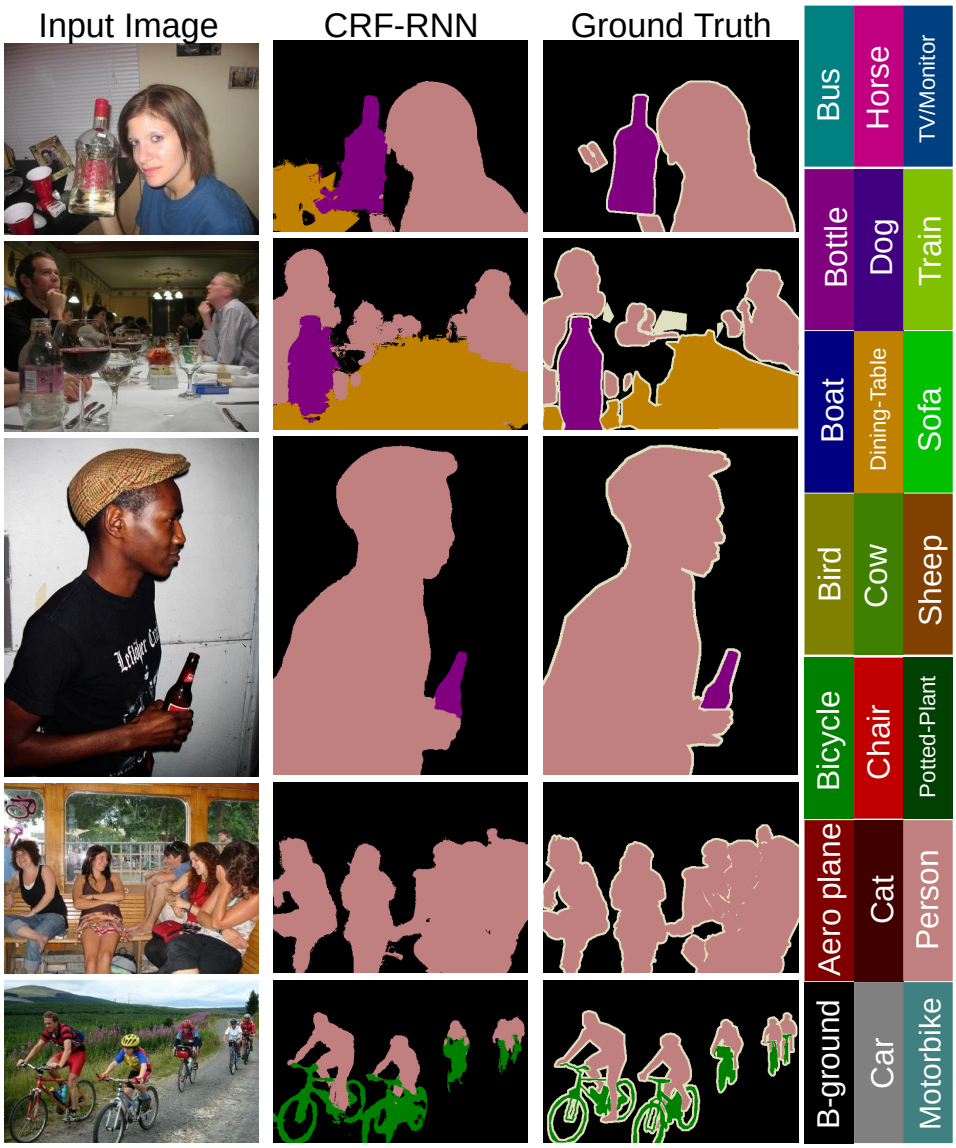


Figure 4.6: **Typical good quality segmentation results I.** Illustration of sample results on the validation set of the Pascal VOC 2012 dataset. Note that in some cases our method is able to pick correct segmentations that are not marked correctly in the ground truth. Best viewed in colour.

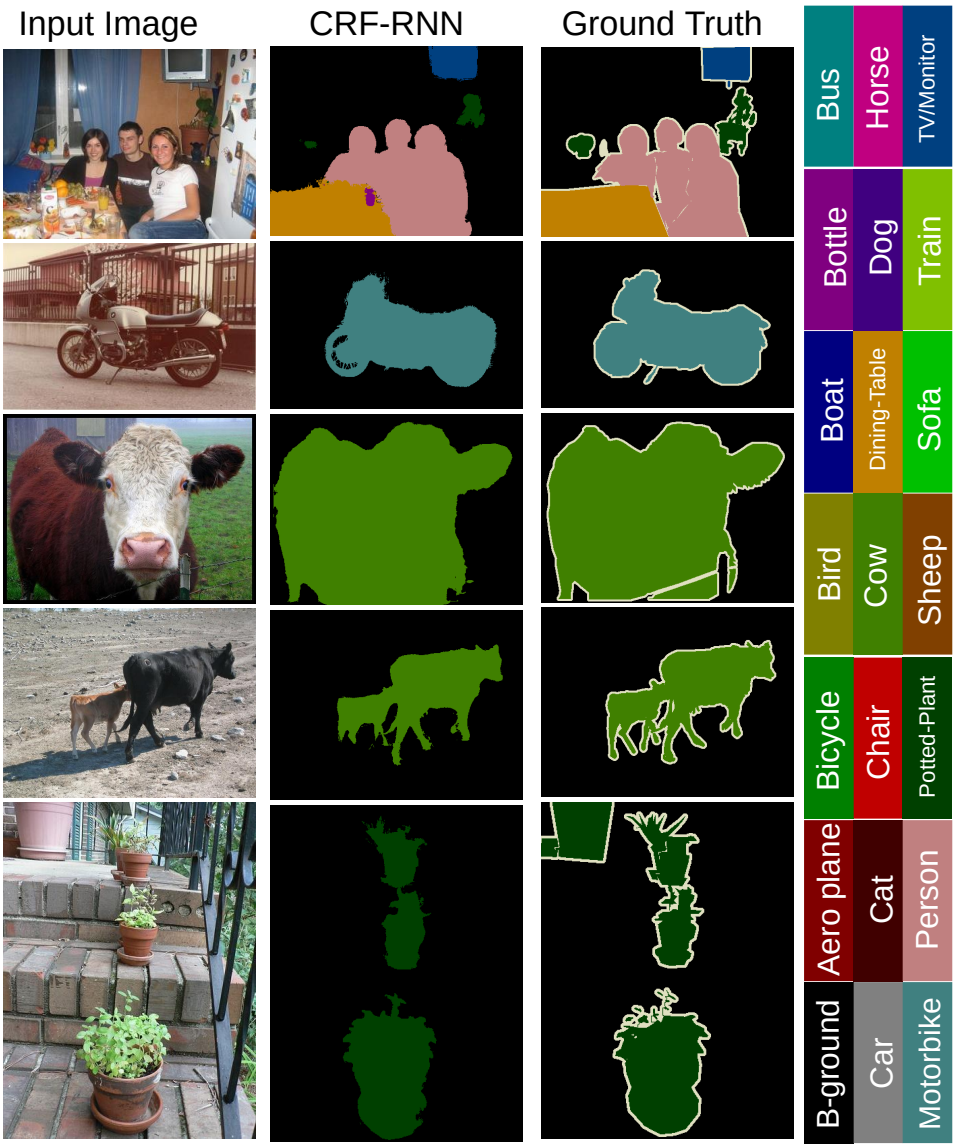


Figure 4.7: **Typical good quality segmentation results II.** Illustration of sample results on the validation set of the Pascal VOC 2012 dataset. Note that in some cases our method is able to pick correct segmentations that are not marked correctly in the ground truth. Best viewed in colour.

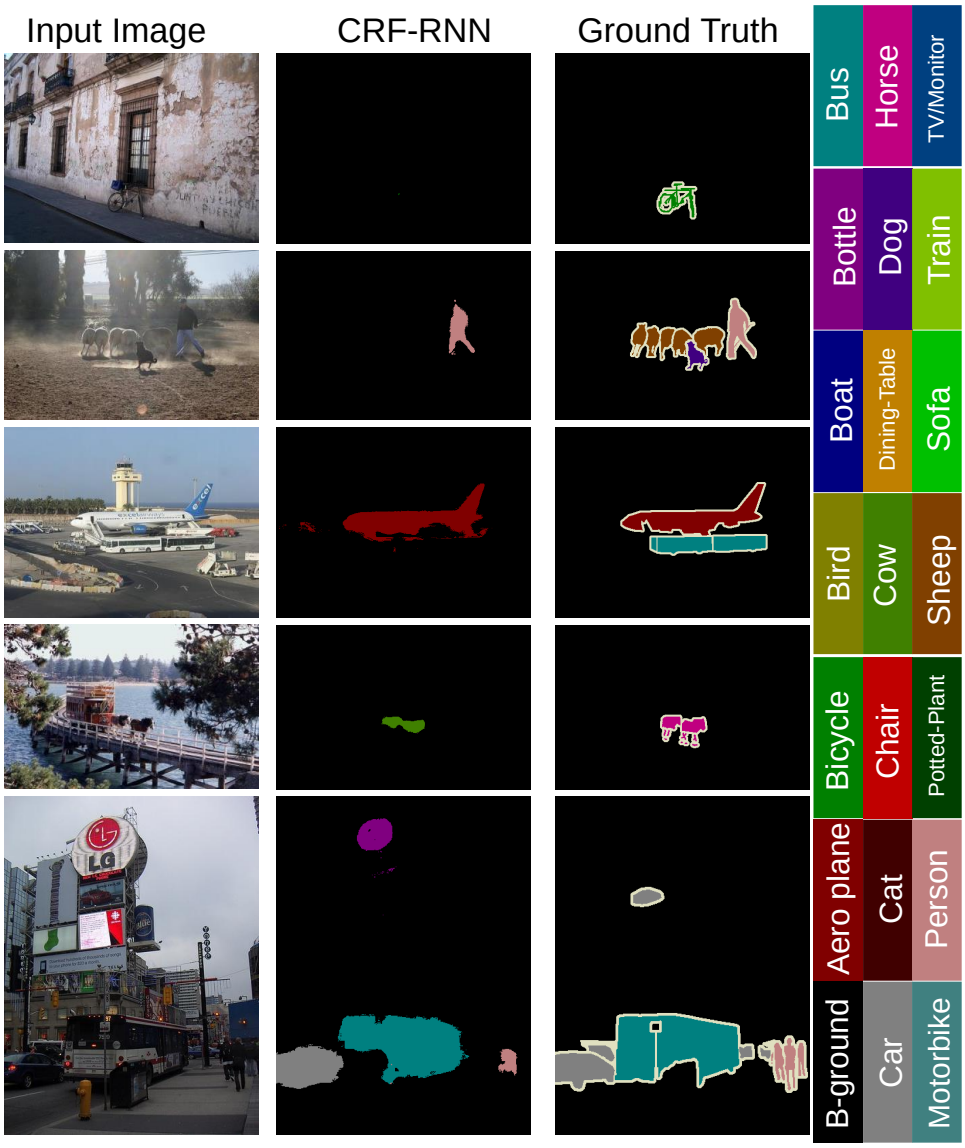


Figure 4.8: **Failure cases I.** Illustration of sample failure cases on the validation set of the Pascal VOC 2012 dataset. Best viewed in colour.

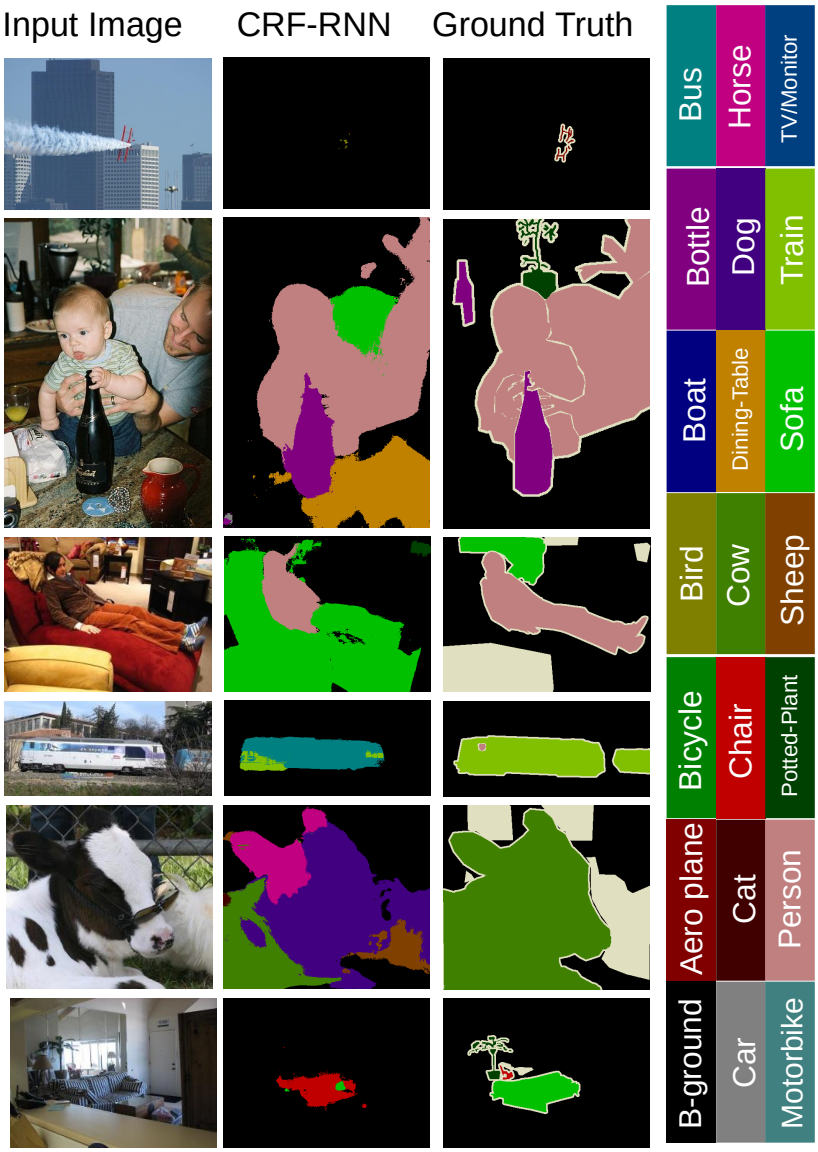


Figure 4.9: **Failure cases II.** Illustration of sample failure cases on the validation set of the Pascal VOC 2012 dataset. Best viewed in colour.

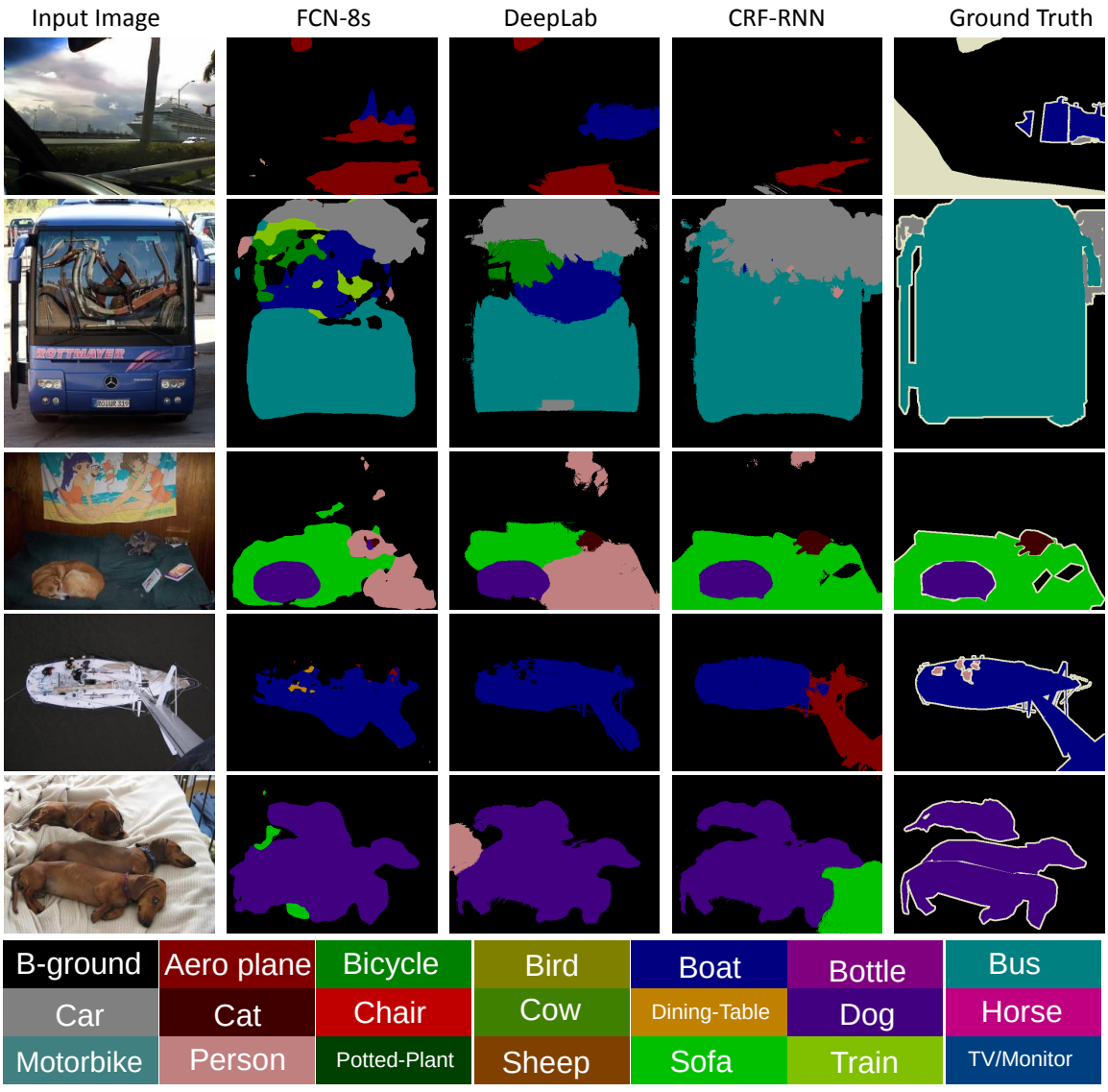


Figure 4.10: **Qualitative comparison with the other approaches.** Sample results with our method on the validation set of the Pascal VOC 2012 dataset, compared with previous state-of-the-art methods. Segmentation results with DeepLap approach were reproduced from the original publication. Best viewed in colour.

Chapter 5

DenseCut: Densely Connected CRFs for Realtime GrabCut

Figure-ground segmentation from bounding box input provided either automatically or manually, has been popular in the last decade and influenced various applications. A lot of research has focused on high-quality segmentation, using complex formulations which often lead to slow techniques, and often hamper practical usage. In this chapter, we demonstrate a very fast segmentation technique which still achieves very high-quality results. We propose to replace the time consuming iterative refinement of global colour models in traditional GrabCut formulation by a densely connected CRF. To motivate this decision, we show that a dense CRF implicitly models unnormalized global colour models for foreground and background. Such relationship provides insightful analysis for bridging between dense CRF and GrabCut functional. We extensively evaluate our algorithm using two important benchmarks. Our experimental results demonstrated that the proposed algorithm achieves an order of magnitude (10X) speed-up on the closest competitor, and at the same time achieves a considerably higher accuracy.

5.1 Introduction

Figure-ground image segmentation from bounding box input, provided either automatically [32, 46, 42] or manually [178], has been extremely popular in the last decade and influenced various computer vision and computer graphics applications, including image editing [120, 45], object detection [189], image classification [228], photo composition [37, 38], scene understanding [108], automatic object class discovery [246], and fine-grained categorization [32]. In order to achieve high quality results, recent methods have focused on complex formulations [220, 126, 203], which typically lead to slow techniques.

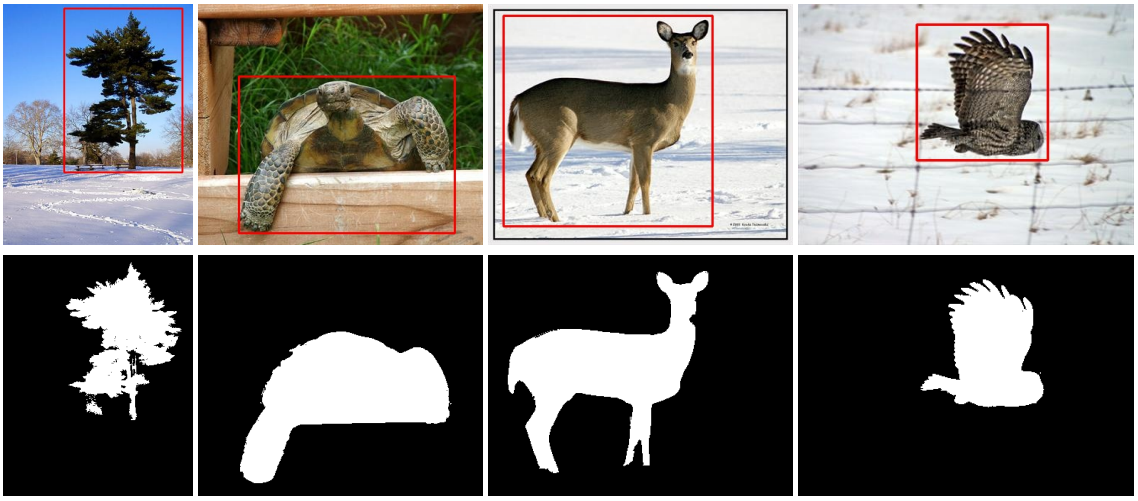


Figure 5.1: Given an input image and a bounding box input (first row), our DenseCut algorithm can be used to produce high quality segmentation results (second row) at real time.

In this work, we aim to design a very fast figure-ground image segmentation technique which still achieves high quality results. We observe that a dense CRF implicitly models an unnormalized global colour model, which is similar to the ones used in the well-known GrabCut functional [178]. We show empirically that the "un-normalization" is not critical in practice. Moreover, we are, to the best of our knowledge, the first to draw a close relationship between dense CRFs and the GrabCut functional. This has surprisingly gone unnoticed by the computer vision community, and yet we believe it to be an interesting result unifying two strands of research on segmentation that provides a deeper insight into the success of the mean field based approach. Given this relationship, we can optimize a densely connected CRF, for which very efficient inference techniques have been recently developed [110], instead of running a slow, iterative refinement of global colour models as in [178], or even slower techniques from [220].

As demonstrated in Fig. 5.1, our algorithm is able to produce high quality figure-ground segmentation results at realtime. To quantitatively evaluate our method against other alternative approaches, we follow recent advances in GrabCut segmentation [203], and extensively evaluate our method on two standard benchmarks, the GRABCUT dataset [178] and the MSRA1K dataset [2] datasets, containing 50 and 1000 images, respectively, with corresponding binary segmentation masks. Our formulation achieves $F_\beta = 93.2\%$ and $F_\beta = 95.9\%$ on the GRABCUT dataset [178] and the MSRA1K dataset [2] dataset respectively, where the F_β represents the harmonic mean of precision and recall. Along with generating better segmentations, our method enables real-time CPU processing which is about $10\times$ faster than its closest competitor [203].

5.2 Related work

Here we review related work that performs interactive figure-ground segmentation [26, 179]. Among the many different approaches proposed over the years, the most successful technique incorporates a per-pixel appearance model and pairwise consistency constraints [21], and uses graph cut for efficient energy minimization [25].

Rother *et al.* [178] proposed the first bounding box based segmentation system that optimised both the appearance model and the segments, using initial appearance models computed from a given bounding box. It was shown by Vicente *et al.* [220] that it is possible to reformulate the GrabCut energy functional [178] in closed form as a higher order MRF, by maximizing over global appearance parameters. This was possible by switching from a Gaussian Mixture Model (GMM) to a histogram representation for the appearance model. However, the optimization of the higher-order MRF is unfortunately NP-hard. Nevertheless, the proposed dual decomposition technique is able to achieve globally optimality in about 60% of cases.

Recently, One Cut [203] by Tang *et al.* has derived a similar formulation. They argue, however, that the part of the higher-order MRF that make the problem NP-hard, *i. e.* the "volume regularization term", is not relevant in practical applications. Hence, they replace this term with a simply unary term, which prefers foreground over background, and can guarantee a globally optimal solution. It is interesting to note that on an abstract level our work has the same line of reasoning. We show that the GrabCut functional and a densely connected CRF formulation are the same under some approximation. We then argue, and demonstrate experimentally, that this approximation is not critical in practice. Training based segmentation methods, *e.g.* "Boxsup" [53] and "CRFasRCNN" [241], have becoming quite popular recently. These methods leverage a carefully trained deep neural networks [98, 195, 143] for high quality sematic segmentation. While these methods are suitable for offline segmentation, the heavy computational overhead makes them unsuitable for realtime interactive applications.

5.3 Methodology

We formulate the figure-ground segmentation problem as a binary label Conditional Random Field (CRF) problem. A CRF is a form of Markov Random Field (MRF) that defines the posterior probability directly, i.e. the probability of the output variables given the input data [20]. The CRF is defined over the random variables $\mathbf{X} = \{X_1, X_2, ..., X_n\}$, where each $X_i \in \{0, 1\}$, 0 for background and 1 for foreground, represents a binary label of the

pixel $i \in \mathcal{N} = \{1, 2, ..., n\}$ such that each random variable corresponds to a pixel. We denote with $\mathbf{x}$ a joint configuration of these random variables, given an observed image data. Based on the general formulation in [110], a fully connected binary label CRF can be defined as:

$$E(\mathbf{x}) = \sum_{i \in \mathcal{N}} \psi_i(x_i) + \sum_{i<j} \psi_{ij}(x_i, x_j), \tag{5.1}$$

where i and j are pixel indices, ψ_i and ψ_{ij} are unary (see Section 5.3.1) and pairwise (see Section 5.3.2) potentials respectively.

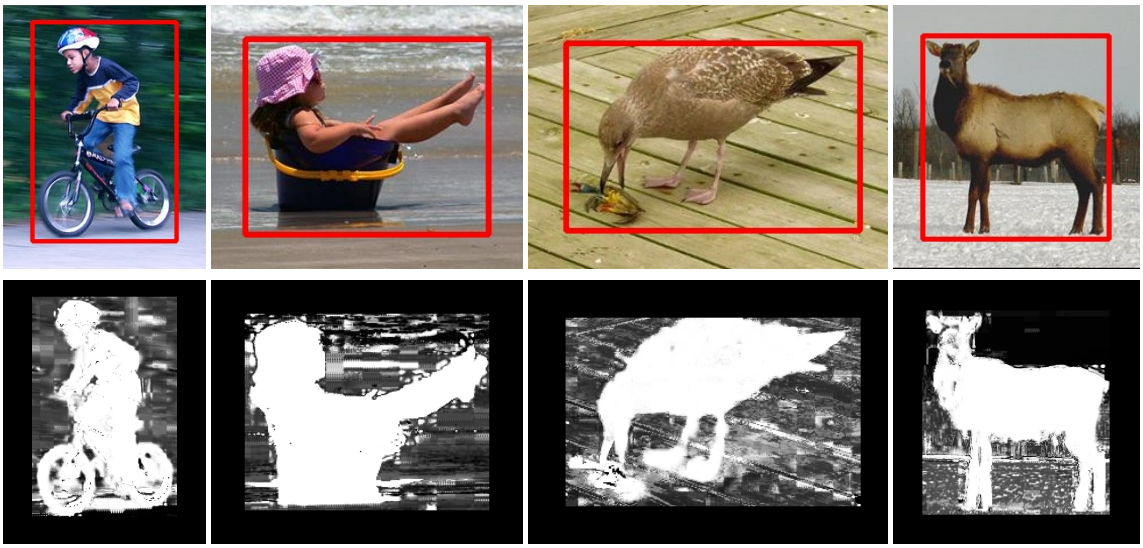


Figure 5.2: Illustration of the probability of each pixel belonging to foreground colour models: sample images and their corresponding $P(x_i = 1)$ are shown in the first and second row respectively.

5.3.1 Unary term estimation

The unary term $\psi_i(x_i)$ measures the cost of assigning a binary label x_i to the pixel i, defined as,

$$\psi_i(x_i) = -\log Pr(x_i), \tag{5.2}$$

which can be computed independently for each pixel by a classier that produces a distributing over the label assignment x_i. Following [131, 169], we use the foreground/background term of the form $Pr(x_i) = \frac{Pr(\Theta_{x_i}, I_i)}{Pr(\Theta_0, I_i) + Pr(\Theta_1, I_i)}$, where $Pr(\Theta_0, I_i), Pr(\Theta_1, I_i) \in (0, \infty)$ represent the probability density value of a pixel colour I_i belonging to the background colour model Θ_0 and the foreground colour model Θ_1, respectively. We use GMMs and follow the implementation details of [202] to estimate the probability density values $Pr(x_i)$ according to the user selection. Examples of $Pr(x_i = 1)$ could be found in Fig. 5.2.

5.3.2 Fully connected pairwise term

The pairwise term ψ_{ij} encourages similar and nearby pixels to take consistent labels. We use a contrast sensitive three kernel potential:

$$\psi_{ij} = g(i,j)[x_i \neq x_j], \tag{5.3}$$

$$g(i,j) = w_1 g_1(i,j) + w_2 g_2(i,j) + w_3 g_3(i,j) \tag{5.4}$$

where the Iverson bracket $[\cdot]$ is 1 for a true condition and 0 otherwise, and the similarity function (5.4) is defined in terms of colour vectors I_i, I_j and position values p_i, p_j:

$$g_1(i,j) = \exp\left(-\frac{|p_i - p_j|^2}{\theta_\alpha^2} - \frac{|I_i - I_j|^2}{\theta_\beta^2}\right), \tag{5.5}$$

$$g_2(i,j) = \exp\left(-\frac{|p_i - p_j|^2}{\theta_\gamma^2}\right), \tag{5.6}$$

$$g_3(i,j) = \exp\left(-\frac{|I_i - I_j|^2}{\theta_\mu^2}\right). \tag{5.7}$$

Here, (5.5) models the appearance similarity and encourages nearby pixels with similar colour to have the same binary label. (5.6) encourages smoothness and helps to remove small isolated regions. The degree of nearness, similarity, and smoothness are controlled by $\theta_\alpha, \theta_\beta$, θ_γ and θ_μ. Intuitively, $\theta_\alpha \gg \theta_\gamma$ should be satisfied if the term (5.5) manages the long range connections and the term (5.6) measures the local smoothness. We use empirical values of $w_1 = 6, w_2 = 10, w_3 = 2, \theta_\alpha = 20, \theta_\beta$ = 33, $\theta_\gamma = 3$ and $\theta_\mu = 43$ in all the experiments of this chapter.

5.3.3 Implementations

Colour modelling: GMMs vs. Histograms: Effective colour modelling is very important for good segmentation results. Among many different models suggested in the literature, two of the most popular ones are histograms [26] and Gaussian Mixture Models (GMMs) [21, 178]. Some important recent works use histogram [203, 220] representations.

In [220], the authors suggest that the MAP estimation with the GMM model is in effect an ill-posed problem, since fitting a Gaussian to the colour of a single pixel may result in an infinite likelihood (see [19]). As explained in [179], this can be avoided by adding a small constant to the covariance matrix. Compared to histograms, GMMs can better adapt to the colours of the image, while still being effective at capturing small appearance differences between foreground and background. Furthermore, the histogram representation will treat different colours equally differently, ignoring the colour values of the histogram

bins, *e.g.* two pixels of a banana might have slightly different colour and be quantised to different bins, even if they are different from the background, with typically a much larger colour difference. We experimentally verify the above discussion via extensive evaluations in Section 5.5.1.

Efficient GMM estimation: As in both the OpenCV [27, 28] and Nvidia CUDA implementation [156], typical GMM estimation can be computationally expensive, due to the large amount of data samples (pixels) used to train the GMMs. In the salient object detection community, more efficient GMM estimation methods have recently been developed [44]. The estimation is made more efficient using an intermediate histogram based representation. Since natural images typically cover a very small portion of all possible colours, uniformly quantizing the image colours (*e.g.* with each channel divided into 12 parts) and then choosing the most frequent colour bins until 95% of image pixels are covered, typically results in a small histogram (*e.g.* an average of 85 histogram bins has been reported [46, 41] for the MSRA1K dataset [2] benchmark). Instead of using hundreds of thousands of image pixels to train the GMM, we can use this small number of histogram bins as weighted samples to train the colour GMM, enabling efficient GMM estimation.

Efficient CRF inference: Our CRF formulation satisfies the general form of the fully connected pairwise CRF with Gaussian edge potentials [110]. This enables us to use highly efficient Gaussian filtering [3] to perform message passing in the mean field framework. Instead of computing the exact Gibbs distribution:

$$P(\mathbf{X}) \propto \exp\left(-E(\mathbf{x})\right) \tag{5.8}$$

of the CRF, we can find a mean field approximation $Q(X)$ of the true distribution $P(\mathbf{X})$, that minimizes the KL-divergence $\mathbf{D}(Q||P)$ among all distributions Q that can be expressed as a product of the independent marginal, $Q(\mathbf{X}) = \Pi_i Q_i(X_i)$ [109]. Minimizing the KL-divergence, while constraining $Q(\mathbf{X})$ and $Q(X_i)$ to be valid distributions, yields the following iterative update equation:

$$Q_i(x_i = l) = \frac{1}{Z_i} \exp\left(\sum_{j \neq i} g(i,j) Q_j(l') - \psi_i(x_i)\right), \tag{5.9}$$

where $l, l' \in \{0, 1\}$, $l' = 1 - l$ are binary variables, and $\frac{1}{Z_i}$ is a normalization factor to constrain $Q(x_i)$ be valid distribution. Each $Q(x_i)$ can be initialized using $Q(x_i) \leftarrow \frac{1}{Z_i} \exp\left(-\psi_i(x_i)\right)$ and then updated using (5.9) until convergence [110]. The final label of each pixel is $\arg\max_{l \in \{0,1\}} Q(x_i = l)$, *i. e.* $Q(x_i = 1) > Q(x_i = 0)$ implies x_i is a foreground pixel.

Naive estimation of the above equation for all image pixels have a high computational complexity, which is quadratic in the number of pixels. We can rewrite the last term of (5.9) by adding and then subtracting $Q_i(l')$ so that

$$\sum_{j \neq i} g(i,j) Q_j(l') = \sum_{j \in \mathcal{N}} g(i,j) Q_j(l') - Q_i(l') \tag{5.10}$$

where $\sum_{j \in \mathcal{N}} g(i,j) Q_j(l')$ is essentially a Gaussian filter, whose value for all image pixels can be calculated efficiently using fast filtering techniques (*e.g.* [109, 110]). This reduces the complexity of the mean field inference, enabling it to be linear to the number of pixels.

5.4 Relationship between fully connected CRF and GrabCut functional

In many figure-ground segmentation methods, *e.g.* GrabCut [178], two (foreground and background) global colour models are explicitly used. Each colour model is derived from its respective region label. This coupling between the pixel labelling and the global colour model leads to a very challenging optimisation, since both parts need to be inferred jointly. In GrabCut this is done in an iterative fashion, while [220] uses dual decomposition. However, both the iterative and dual decomposition optimisations are slow, with the latter taking up to minutes per frame.

In this work we replace the global colour model with a single optimization of a fully connected CRF. This is based on the insight that a fully connected CRF and a standard low-connected (*e.g.* 8-connected) CRF with associated foreground and background global colour models are very closely related, in the sense that the former is an approximation of the latter. This approximation is nearly exact when the area of the fore- and background region is the same in the final segmentation. In the following we also draw a relationship to the One Cut [203] work, since the approximations in their work and ours are related.

This observation suggested that we can avoid the computational expensive process of global colour model estimation, and use the efficient inference for fully connected CRF to enable very fast computation.

Let us consider a specific form of our fully connected CRF, where $w_2 = 0$. Note that this is only a minor change to the energy (5.1) since the spatial smoothness term is still present in g_1. The energy is then given as

$$E(\mathbf{x}) = E_1(\mathbf{x}) + w_3 \sum_{i<j} g_3(i,j)[x_i \neq x_j], \tag{5.11}$$

$$E_1(\mathbf{x}) = \sum_{i \in \mathcal{N}} \psi_i(x_i) + w_1 \sum_{i<j} g_1(i,j)[x_i \neq x_j]. \tag{5.12}$$

Let us now write the Grabcut functional as given in [178]

$$E(\mathbf{x}, \Theta_B, \Theta_F) = \sum_{i \in \mathcal{N}} (P_B(I_i; \Theta_B)[x_i = 0] + P_F(I_i; \Theta_F)[x_i = 1]) + \sum_{(i,j) \in \mathcal{N}_8} \frac{1}{|p_i - p_j|^2} \exp(-\beta |I_i - I_j|^2)[x_i \neq x_j]. \tag{5.13}$$

Here Θ_F and Θ_B are the foreground and background Gaussian mixture models respectively, $P_F(I_i; \Theta_F)$ and $P_B(I_i; \Theta_B)$ are the negative log probability of the colour I_i under the respective Gaussian mixture model. The second summand represents the popular edge-preserving smoothing term, here over an 8-Neighborhood grid, and β is a constant defined in [178]. Note, we are interested in the minimizer $\mathbf{x}^* = \arg\min_{\mathbf{x}} \min_{\Theta_F, \Theta_B} E(\mathbf{x}, \Theta_B, \Theta_F)$.

One difference between (5.11) and (5.13) is that the unary term is missing, i.e. $\sum_{i \in \mathcal{N}} \psi_i(x_i)$, in (5.13). Furthermore, let us show that the edge-preserving smoothing term in (5.13) is very similar to g_1. This can be seen by re-writing the second summand as:

$$\sum_{(i,j) \in \mathcal{N}_8} \frac{1}{|p_i - p_j|^2} \exp(-\beta |I_i - I_j|^2)[x_i \neq x_j] = \sum_{(i,j) \in \mathcal{N}_8} \exp(-\log |p_i - p_j|^2 - \beta |I_i - I_j|^2)[x_i \neq x_j]. \tag{5.14}$$

If you compare this equation with (5.5) then the first difference is the "log" operator for the pixel distance. The second difference is that we have an 8-neighborhood system instead of a fully connected system. However, by choosing θ_α and θ_β accordingly this can approximated.

Let us now define a version of GrabCut, with a slightly modified edge-preserving smoothing as

$$E(\mathbf{x}, \Theta_B, \Theta_F) = E_1(\mathbf{x}) + \sum_{i \in \mathcal{N}} (P_B(I_i, \Theta_B)[x_i = 0] + P_F(I_i; \Theta_F)[x_i = 1]). \tag{5.15}$$

The only difference between the GrabCut function and the fully connected CRF is the term g_3 in (5.11) and the sum over the negative log probability in (5.15).

Let us define the following function that computes a distance between a colour, here I_i, and distribution of colours, here all colours of the background region:

$$P'_B(I_i) = \frac{1}{|\mathcal{N}_B|} \sum_{j \in \mathcal{N}_B} K(I_i, I_j) \tag{5.16}$$

$$\text{with kernel: } K(I_i, I_j) = -\frac{1}{2} \exp\left(\frac{-|I_i - I_j|^2}{2\theta_\mu^2}\right), \tag{5.17}$$

		MSRA1K dataset [2]		GRABCUT dataset [178]	
		F_β measure	Time (s)	F_β measure	Time (s)
CPU	GrabCut [178]	0.945	1.22	0.909	2.02
	One Cut [203]	0.949	0.664	0.900	1.70
	Ours	0.959	0.075	0.932	0.143
CUDA	GrabCut(GMM) [156]	0.949	0.074	0.918	0.149
	GrabCut (Histogram)[156]	0.889	0.059	0.714	0.135

Table 5.1: Average precision, recall, F_β, and processing time (measured in seconds) on two well known benchmarks (see Fig. 5.3 for sample results). Tested on a computer with Intel Xeon E5645 2.40GHz CUP, 4GB RAM, Nvidia Tesla K40 GPU and CUDA 7.0 SDK.

where $\mathcal{N}_B$ is the set of background pixels, *i. e.* $x_i = 0$. Note that this can be seen as a Parzen-Density estimator with an infinite support region. In essence, $P'_B(I_i)$ is the average kernel-distance of the colour I_i at pixel i with all colours that are assigned to background. The equivalent distance estimator for foreground is defined as: $P'_F(I_i) = \frac{1}{|\mathcal{N}_F|} \sum_{j \in \mathcal{N}_F} K(I_i, I_j)$.

We can now state the following theorem that relates the GrabCut function in (5.15) with our fully connected CRF in (5.11).

Theorem 5.4.1. Two minimizers $\arg\min_{\mathbf{x}} E(\mathbf{x})$ of (5.11) and $\arg\min_{\mathbf{x}} \min_{\Theta_F, \Theta_B} E(\mathbf{x}, \Theta_F, \Theta_B)$ of (5.15) are the same if we replace the global colour-model functions $P_F(I_i; \Theta_F)$ and $P_B(I_i; \Theta_B)$ in (5.15) by weighted functions $|\mathcal{N}_F| P'_F(I_i)$ and $|\mathcal{N}_B| P'_B(I_i)$, respectively.

Proof. Let us look at the function $\sum_{i<j} g_3(i,j)[x_i \neq x_j]$, which is part of (5.11) but not

(5.15). The minimizer for the function can be re-written as follows:

$$\arg\min_{\mathbf{x}} \sum_{i<j} g_3(i,j)[x_i \neq x_j] \tag{5.18}$$

$$= \arg\min_{\mathbf{x}} \sum_{i<j} g_3(i,j)[x_i \neq x_j] - \sum_{i<j} g_3(i,j)$$

$$= \arg\min_{\mathbf{x}} \sum_{i<j} -g_3(i,j)[x_i = x_j]$$

$$= \arg\min_{\mathbf{x}} \sum_{i\in\mathcal{N}} (\sum_{j\in\mathcal{N}} -\frac{1}{2} g_3(i,j)[x_i = x_j])$$

$$= \arg\min_{\mathbf{x}} \sum_{i\in\mathcal{N}} (\sum_{j\in\mathcal{N}_B} K(I_i, I_j)[x_i = 0] +$$

$$\sum_{j\in\mathcal{N}_F} K(I_i, I_j)[x_i = 1]) \tag{5.19}$$

$$= \arg\min_{\mathbf{x}} \sum_{i\in\mathcal{N}} (|\mathcal{N}_B| P'_B(I_i)[x_i = 0] +$$

$$|\mathcal{N}_F| P'_F(I_i)[x_i = 1]). \tag{5.20}$$

Comparing (5.20) and (5.15) shows the demanded relationship. □

The remaining question is: What is the effect of the "weighting" of the functions $P'_F(I_i)$ and $P'_B(I_i)$? First of all, observe that we would ideally like to get rid of the weights $|\mathcal{N}_F|$ and $|\mathcal{N}_B|$, since this would give us a proper (infinite) Parzen-window estimator. However, intuitively this is not possible since [220] has shown that solving the GrabCut function is NP-hard. We call this approximation, *i. e.* $|\mathcal{N}_F| P'_F(I_i)$ instead of $P'_F(I_i)$ the "unnormalized global colour model". It can be seen that if the ratio $\frac{|\mathcal{N}_F|}{|\mathcal{N}_B|} = 1$ then we actually have a proper density estimator, since all weights can be globally re-scaled. This means that we can compute the global minimizer $\mathbf{x}$ for (5.11) and analyze its ratio. If the ratio is close to 1, it means that it is close to a proper density estimation. By choosing a rectangle image region outside the bounding box input as a working region to build CRF, we can roughly control this ratio. In our experiments, we select a $w_b = 5$ pixel wider region than the bounding box input as working region, which generates an average ratio of 1.5 and 1.2 for MSRA1000 and GRABCUT benchmarks, respectively. We experimentally find that changing w_b in a large range, *e.g.* $[2, 10]$, has an negligible influence on the algorithm performance.

It is interesting to note that this discussion is related to the main line of argumentation in the One Cut [203] work. In One Cut [203], the authors re-write the GrabCut functional by replacing the "volume regularization term" with a simple ballooning force (unary term) that prefers to have all pixels being foreground. This change makes it possible to optimize

the new GrabCut functional globally optimal. The "volume regularization term" enforces that segmentations with a ratio $\frac{|\mathcal{N}_F|}{|\mathcal{N}_B|} = 1$ are preferred, *i. e.* it penalizes segmentations with extreme ratios. They observe empirically that removing this regularization term does not affect results. In the above discussion we also derived a theoretically sound method for the case that $\frac{|\mathcal{N}_F|}{|\mathcal{N}_B|} = 1$. However, as in [203], ignoring this ratio constraint gives us good results in practice.

5.5 Experiments

We extensively evaluate our method on two well known benchmarks (MSRA1K dataset [2] and GRABCUT dataset [178]), and compare our results with the state-of-the-art alternatives [178, 203], in terms of segmentation quality and efficiency.

5.5.1 Segmentation Quality Comparison

We evaluate the binary segmentation performance of each method given a user bounding box around the object of interest. The GRABCUT dataset [178] benchmark contains 50 images with bounding box and binary mask annotations. For MSRA1K dataset [2] benchmark, we export the bounding box annotation from its binary mask ground truth, and use this bounding box as input to each method.

To objectively evaluate our method, we compare our results with the two other state-of-the-art methods for bounding box-based figure-ground segmentation *i. e.* GrabCut [178] and One Cut [203]. For GrabCut, we use the CPU implementation from OpenCV [28] and two highly optimised commercial GPU implementations from Nvidia [156] (one uses a GMM colour model and another one uses a histogram colour model). Average precision, recall, and F-Measure are compared against the entire ground truth datasets, with F-Measure defined as harmonic mean of precision and recall:

$$F_\beta = \frac{(1+\beta^2)\text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}. \tag{5.21}$$

Table 5.1 shows the average precision, recall, and F_β values (we use $\beta^2 = 0.3$ as in [2, 46, 203]). Visual examples of input bounding boxes and segmentation results are shown in Fig. 5.3. Among the baseline methods, the commercial GPU GrabCut implementation from Nvidia [156] achieves the best segmentation results. Although faster computationally, the histogram representation has limited ability to precisely capture appearance differences, resulting in significantly worse segmentation results than the GMM based representation. The comparison between the two versions of Nvidia's commercial implementation clearly

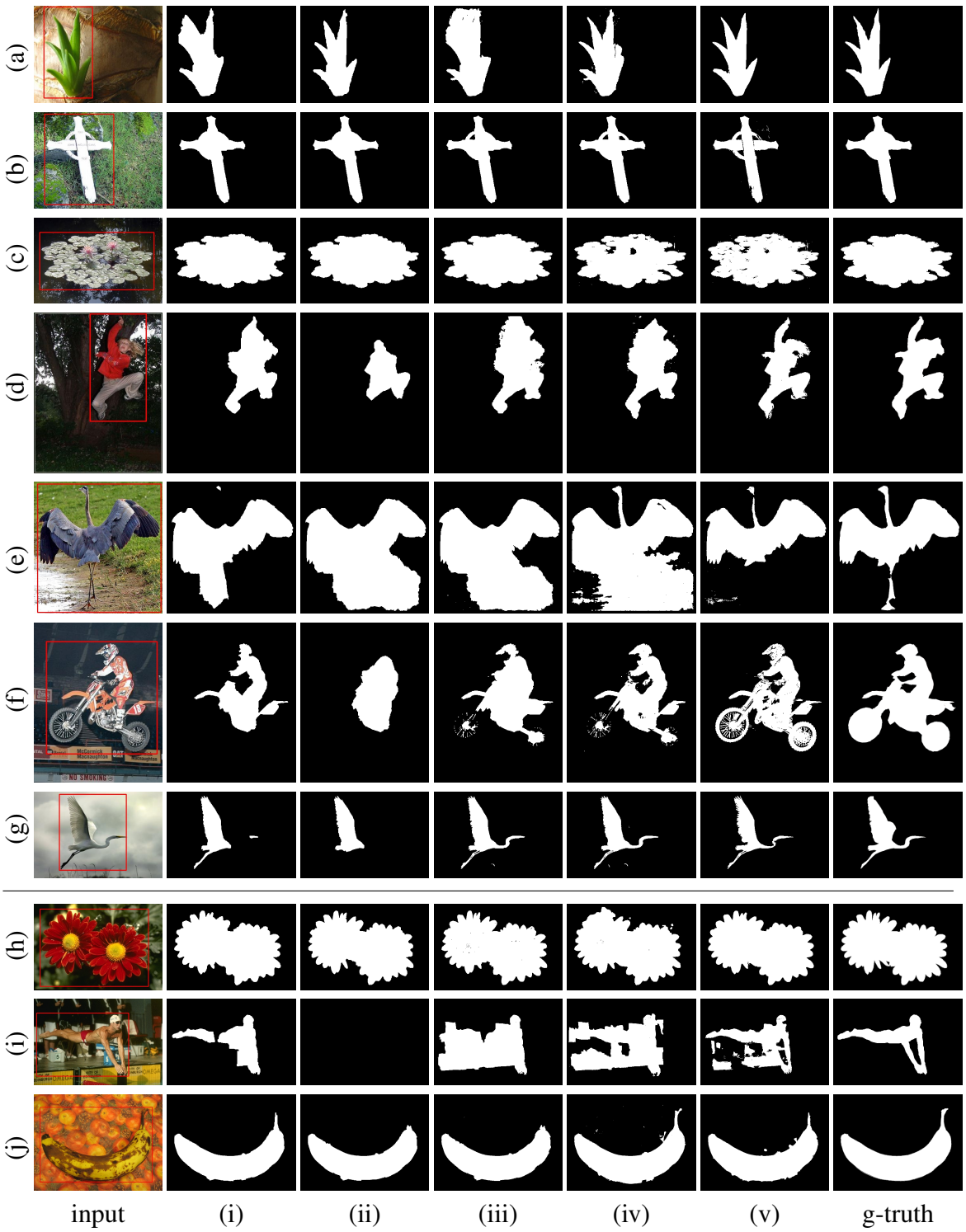
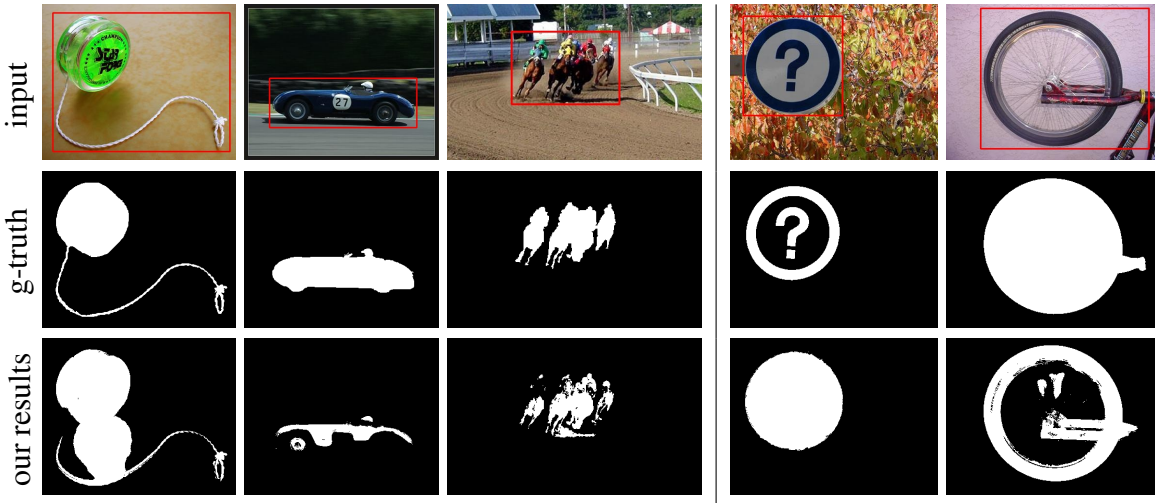


Figure 5.3: Sample results for images from MSRA1K dataset [2] (a-g) and GRABCUT dataset [178] (h-j) benchmarks, using different methods: (i) GrabCut [156]GMM, (ii) GrabCut [156]Hist., (iii) GrabCut [178], (iv) One Cut [203], and (v) Ours.



(a) g-truth is of high quality (b) g-truth is of low quality

Figure 5.4: Examples for top 50 'failing examples' shows that our results are very often comparable to ground truth annotations: (a) ground truth mask in MSRA1000 benchmark [2] is preferred, (b) our segmentation results is preferred.

verifies our discussion in Section 5.3.3. In both the benchmarks, our method consistently produces better segmentation results than all other alternatives.

While we have shown theoretically that GrabCut, One Cut and our Dense CRF are very related, we believe that these differences in performance stem from the fact that we have more parameters to adjust. Hence the weighting between the kernels that relate to spatial smoothing, contrast based smoothing, and global colour models, are more finely tuned. This is noticeable visually - see for instance the fine details of the target object regions that are successfully segmented in Fig. 5.3(c) and Fig. 5.3(f) by our method.

Comparing One Cut with our method, we notice that, on average, our method produces better results than One Cut, possibly due to the more powerful colour model representation. Extending the One Cut method to incorporate GMMs for representing colours is non-trivial and known to be a NP-hard problem [203, 220].

Due to explicitly enforcing colour separation between foreground and background, only One Cut provides results similar to our own. Both methods recover more accurate fine object boundaries than the other methods, *e.g.* Fig. 5.3(c)(d)(f).

5.5.2 Computational time

As shown in Table 5.1 our method is about $10\times$ faster than any other current CPU based implementation. Implementing a GPU version to fully explore the parallel nature of the algorithm is a promising direction for future work.

Due to the use of the very efficient GMM representation of [44], the most computationally expensive part of our algorithm is the mean field based inference [110], which could be efficiently solved using advanced bilateral filtering techniques [3]. It is worth mentioning that the mean field based inference is an intrinsically parallel algorithm, and thus can be made further efficient using graphics hardware (GPU) or multi-core CPUs. In our current implementation we use OPENMP instructions to parallelize across multiple CPU cores.

5.5.3 Limitations

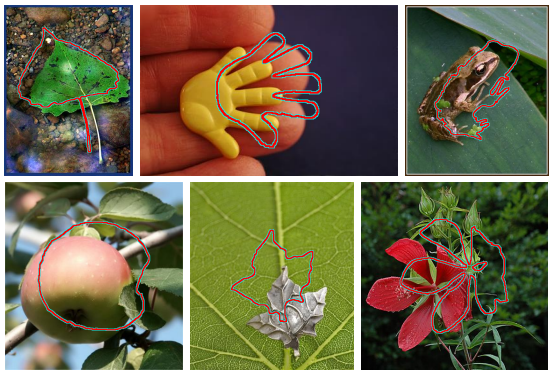


Figure 5.5: We found ground truth errors in the MSRA1000 benchmark [2] as shown above (the red lines on top of each image illustrate the contour of the ground truth mask). After a manual check, we found 9 such errors from all the annotations of 1000 images, all such ground truth errors are found in the top 6% 'failing cases'.

The high accuracy of our method ($F_\beta = 95.9\%$ for the MSRA1K dataset [2] benchmark and $F_\beta = 93.2\%$ for the GRABCUT dataset [178] benchmark), indicates that most results of our methods are very similar to the ground truth. This make it feasible to visualise and study all the clearly failing examples even for a large benchmark such as MSRA1K dataset [2]. We do this by studying the top 50 'failing examples', which are automatically selected as the results with lowest F_β values according to ground truth. We found that the MSRA1K dataset [2] benchmark, although used as standard benchmark for figure-ground segmentation (having currently 1100+ citations), contains some clear ground truth errors as shown in Fig. 5.5 (where ground truth masks appear shifted due to unknown reasons). Note that, besides these errors (less than 1%), which we could easily detect from top 6% 'failing cases', most of the other ground truth annotations are of very high quality.

Fig. 5.4(a) shows typical examples of top 'failing cases'. In the first example, the shadow part occurs only inside the bounding box and its appearance is quite different compared with pixels outside the bounding boxes, forcing the algorithm consider it as an object

region. In the other two failure cases, some foreground regions have a large portion of similar appearance regions outside the bounding box, which confuses the algorithm and leads to missing regions for the target object. We went through top 50 'failing cases' and found 12 cases with low quality ground truth segmentation (see also Fig. 5.4) and 8 cases with incorrect segmentation (see also Fig. 5.5).

5.6 Conclusions

We have presented an efficient figure-ground image segmentation method, which uses fully connected CRF for effective label consistency modelling. Formally, we show that a fully connected CRF, as used in this work, and the well-known GrabCut functional, with a low-connected, *e.g.* 8-connected, CRF with associated foreground and background global colour models are closely related. This motivated us to replace the global colour model in the traditional GrabCut framework with a single optimization of a fully connected CRF. Extensive evaluation on two well known benchmarks, MSRA1K dataset [2] and GRABCUT dataset [178], demonstrates that our methods is able to get more accurate segmentation results compared to other state-of-the-art alternative methods, while achieving an order of magnitude speed-up with respect to the closest competitor.

Further introducing a bounding box prior [126], or high order terms [224] could be useful future additions to our framework.

Chapter 6

Discussion

6.1 Findings

In Chapter 2, we presented an efficient, hierarchical, fully-connected multi-label conditional random field (CRF) framework. This framework addresses the multi-labelling problems such as semantic image segmentation for objects and visual attributes. We also proposed a piecewise boosting-based training strategy to learn the label correlations based on visual appearance similarity and label co-occurrence statistics. We demonstrated that the proposed framework can combine information successfully from visual attributes and objects at region- and pixel- levels in the task of semantic image segmentation. We found that per-pixel visual attribute segmentation contributes to achieving higher accuracy and finer semantic segmentation results. We generalized the fully-connected CRFs with Gaussian pairwise potential for multi-labelling problems by making use of this property of the underling inference algorithm: the approximate marginal distribution is fully factorisable. Following Krähenbühl *et al.* [110], we adopted the filter-based mean-field approximate inference. This inference involves finding an approximate marginal distribution that minimizes the KL-divergence between the actual marginal distribution and the proposed one. Based on Koller & Friedman [109], a product of independent marginals can express this approximate marginal distribution. Given the form of our problem, we can factorize the approximate marginal distribution into a product of marginals over the multi-class object and binary visual attribute variables.

The multi-label CRFs framework was employed in Chapter 3 to develop an interactive image segmentation system. We proposed a system that allows a user to verbally refine the semantic image segmentation results. Based on the multi-label CRF framework, we developed a semantic image segmentation system that can assign both object labels and visual attribute labels to each image pixel. The attribute labels act as verbal handles through

which users can control the CRF, allowing them to refine the semantic image segmentation results. Despite the ambiguity of verbal commands, our system delivered reasonable segmentation results. This hands-free interactive segmentation provides verbal methods to select objects of interest, which can be used to aid image editing applications.

Chapter 4 investigated the connection between the fully-connected CRFs with Gaussian pairwise potentials and recurrent neural networks. We found that an iteration of the filter-based mean-field approximation can be implemented using a series of convolutional neural network atomic operations. Hence we formulated the whole iterative inference process as a recurrent neural network. The interpretation of fully-connected CRFs integrates the CRF-based probabilistic graphical modelling with emerging deep learning techniques. In particular, the proposed CRF-RNN can be plugged in as a part of a deep neural network to achieve an end-to-end trainable system. CRF-RNN allows passing error differentials from its outputs to inputs during back-propagation based training of the deep network while learning the parameters of CRFs. We demonstrated the effectiveness of this approach on the task of semantic image segmentation.

In Chapter 5, we found the relationship between the fully-connected CRFs with Gaussian pairwise potentials and GrabCut [178]. Considering the problem of figure-ground segmentation from bounding box input, we discovered that a fully-connected CRFs with Gaussian pairwise potentials implicitly model the un-normlized global colour models for foreground and background. In GrabCut [178], the two (foreground and background) global colour models are explicitly used. In GrabCut, optimization is done in an iterative fashion, which is considerably slow in a practical system. Based on the relationship we found, we replaced the global colour model with a single optimization of fully-connected CRF. The optimization is then done with the efficient filter-based mean-field approximate inference.

6.2 Limitations

While the proposed segmentation techniques described in chapter 2 and 4 have proved powerful, they are not able to handle well all the appearance variations that images presented. These techniques are based on supervised learning approaches. In particular, these techniques are only trained on certain images *e.g.* images from NYU v2, Pascal VOC, Microsoft COCO datasets. The appearance variations in these images are not necessarily well representing the real-world images. For real-world specific applications such as Google photos and autonomous vehicles, the models pre-trained on standard academic datasets would not generalize well on new types of datasets, since the viewpoint, scales, occlusions, and lighting would vary significantly in different scenes.

The data set collection and ground truth for semantic image segmentation are still burdensome. Without collecting and annotating sufficient amount of high quality data, supervised learning techniques developed in this thesis would be difficult to successfully apply. For some applications such as heath-care research, it is expensive or sometimes impossible to annotate large amount of detailed pixel-wise label maps.

The presented technique described in chapter 3 attempted to overcome this problem by establishing a new dataset with both objects and visual attributes labels. However, this technique still relies on the predefined sets of labels. In many real-world applications such as Google photos, image analysis would require to be able to respond to arbitrary images. These images often contain the object classes that are not very well defined or represented in the predefined label sets.

The proposed techniques described in chapter 2, 3, 4, 5 are based on the filter-based mean-field approximate inference algorithm and fully-connected CRF. We consider this fully-connected CRF because it allows long-range interactions, and long-range interactions [108] was shown to be useful in semantic image segmentation. We restrict the pairwise potential functions to be a weighted sum of Gaussian kernels so that we can make the computations feasible. This restriction allows us to make use of efficient bilateral filters, such as the permutohedral lattice [3]. However, the proposed approach is not efficient for the discrete version of the continuous structured prediction problem on applications such as optical flow estimation and depth estimation, *e.g.* the problem with 256 labels.

The semantic image segmentation techniques proposed in this thesis do not provide more detailed information about the instances of each visual object category. Although it is useful to have per-pixel semantic labels, some applications such as intelligent visual surveillance might require having both per-pixel semantic labels as well as per-pixel instance labels.

6.3 Future Work

To address these limitations, we propose several thoughts for future research.

Generating synthetic data for training a semantic image segmentation system is a promising direction. For the application of autonomous vehicles, training a semantic image segmentation system would require a significant amount of high quality annotated data. However, it is impossible to collect the real data from scenarios such as traffic accidents. By generating synthetic data through computer graphics, we would have full control on the way of generating data. We would then be able to collect the data that happens in the long

tails, e.g. traffic accident scenes. SYNTHIA dataset [176] has demonstrated promising results in this direction.

Transfer learning for semantic image segmentation is another interesting future direction. In fact, the existing state-of-the-art performance on semantic image segmentation for Pascal VOC dataset [241] was achieved by fine-tuning the model that was previously trained on ImageNet dataset. One future direction is to investigate how to efficiently transfer the best semantic image segmentation model trained on Pascal VOC to other datasets and other problems. Chen [39] has demonstrated an efficient knowledge transfer method that produces promising results on ImageNet. Future direction would investigate if this applied for semantic image segmentation.

Structured prediction with large label spaces is a challenging problem. Many practical problems such as optical flow estimation and depth reconstruction are in this category. Recent work [36] demonstrated an efficient inference based on an continous optimization method such as block coordinate descent. This works well on this type of problems such as depth reconstruction, and optical flow [148]. It would be interesting to investigate if variational inference algorithms would work well in this type of problem.

Instance segmentation is a next exciting research direction. Its goal is to delineate visual objects and recognize both its instance identities as well as its category. Current semantic segmentation provides pixel-wise semantic class labels. However, it is not able to distinguish the instances belonging to the same semantic category. This ability is important for many applications like autonomous vehicles and intelligent visual surveillance. Recent work has already demonstrated the promising initial results [88, 40, 89, 54, 133, 129, 55, 140, 237, 174, 8]. Another interesting direction [194, 238, 215] is the problem of instance segmentation based on RGB and depth images, where the information from depth sensors helps to better handle the occlusions.

6.4 Final Remarks

Semantic image segmentation has been significantly advanced after the pioneering works of Duygulu [62] and Shotton [192]. Thanks to deep learning, general-purpose graphics processing units (GPGPUs), and large-scale datasets like Pascal VOC [68] and Microsoft COCO [135], the community has dramatically improved the state-of-the-art performance of visual object recognition [195, 201, 90] and semantic image segmentation [143] over the

last few years. The proposed techniques [241] have demonstrated the promising direction to further improve the performance by integrating deep learning and probabilistic graphical models. Although deep convolutional neural networks have achieved success in many different domains, they have shortcomings, such as lacking of the capability of modeling long-term dependencies [166]. The key insight is to formulate the learning and inference algorithm of probabilistic graphical models in a way that could fully take advantages of the strength of deep learning and the strength of probabilistic graphical models. This insight has also achieved promising results in joint detection and segmentation [7], instance segmentation [8], and image synthesis [128]. Traditional semantic image segmentation works are mostly focusing on learning to recognize visual object categories. To understand and precisely describe the visual objects, it is also important to have fine-grained detailed information about objects such as materials, and surface properties. Some preliminary work in this direction is presented in Zheng *et al.* [240]. The forthcoming Visual genome challenge [112] will also try to push the envelope of achievable detailed visual object recognition.

Appendix A

Filter-based Mean-Field Approximate Inference

A.1 Introduction

The aim of this appendix is to briefly summarize the algorithm of filter-based mean-field approximate inference. Mean-field approximate inference is one important type of variational inference algorithms. It was shown effective in semantic image segmentation and foreground-background segmentation when we combine it with efficient filtering approaches such as bilateral filters.

A.2 Mean-field approximation

For the problem of semantic image segmentation, consider a random field defined over random variables $\mathcal{X} = \{X_1, ..., X_N\}$ that is conditioned on an image. Each random variable is associated with a pixel in the image I, where the set of pixel index is denoted as $\mathcal{N} = \{1, ..., N\}$. We can then define the Gibbs marginal distribution for the problem as follows.

$$P(\mathbf{X}|I) = \frac{1}{Z}\tilde{P}(\mathbf{X}) = \frac{1}{Z}\exp(-E(\mathbf{X})) \tag{A.1}$$

where $E(\mathbf{X})$, $Z = \sum \exp(-E(\mathbf{X}))$ are respectively the energy function associated with the configuration $\mathbf{X}$, and the partition function. Notice that $E(\mathbf{X})$ can also be written as $E(\mathbf{X}|I)$ to reflect that this energy function is conditioned on the input image I [223]. The partition function is defined as $Z = \sum_{\mathbf{X}} \tilde{P}(\mathbf{X})$. The energy function is broken down as follows.

$$E(\mathbf{X}) = \sum_{i \in \mathcal{N}} \psi_u(x_i) + \sum_{i<j \in \mathcal{N}} \psi_p(x_i, x_j), \tag{A.2}$$

where the first term $\psi_u(x_i)$ is the unary potential functions that can take arbitrary form, e.g. Textonboost [192], or the output of a Fully Convolutional Network (FCN) [143]. While the second term corresponds to pairwise potential functions. Long-range interactions was shown improving the semantic segmentation [108]. We would like to take into account the long-range interactions by operating under the fully-connected assumption. To make the computations feasible in practice, we restrict that pairwise potential functions as a weighted sum of Gaussian kernels. This restriction allows us to make use of efficient bilateral filters, such as permutohedral lattice [3].

In order to take advantages of the efficient bilateral filter such as permutohedral lattice [3] and the fully-connected assumption, the pairwise potential functions are defined to take the form of a weighted of Gaussian kernels:

$$\psi_p(x_i, x_j) = \psi(x_i, x_j) \sum_{m=1}^{K} w^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j), \tag{A.3}$$

where the first term $\psi(x_i, x_j)$ is an *arbitrary label compatibility function*, while the functions $k^{(m)}(.,.),, m = 1, ..., M$ are Gaussian kernel functions defined over feature vectors $\mathbf{f}_i, \mathbf{f}_j$. Label compatibility function represents the distance between labels, while the Gaussian kernel functions represent the distance between pixels.

In semantic image segmentation [110, 241], for the Gaussian kernel functions, these feature vectors $\mathbf{f}_i, \mathbf{f}_j$ are derived based on the image pixel data at locations i and j. In particular, Krähenbühl *et al.* [110] defined the form of $\mathbf{f}_i$ by concatenating the intensity values at pixel i with the horizontal and vertical positions of pixel i in the image. $w^{(m)}, m = 1, ..., M$ are applied to weight the kernels.

Let the approximate marginal distribution be defined as $Q(\mathbf{X})$. We assume that this approximate marginal distribution is fully factorisable, meaning that we can represent the approximate marginal distribution as a product of independent marginals over X_i, $Q(\mathbf{X}) = \Pi_i Q_i(X_i)$.

The KL-divergence measures the distance between the approximate marginal distribution Q and the true one P. Let us refer $\mathbf{E}_{\mathbf{X}\sim Q}$ to the expected value under the distribution Q. Given equation A.1, we have $\log P(\mathbf{X}) = \log \tilde{P}(\mathbf{X}) - \log Z = -E(\mathbf{X}) - \log Z$. We also take into account the assumption that the approximate marginal distribution can be factorized into a product of independent margins over X_i. Due to the linearity of expectation [110], Shannon entropy decomposes $\mathbf{E}_{\mathbf{X}\sim Q}[\log Q(\mathbf{X})] = \sum_i \mathbf{E}_{X_i\sim Q}[\log Q(X_i)]$ when $Q(\mathbf{X}) = \Pi_i Q_i(X_i)$. We can then rearrange the form KL-divergence as follows.

$$
\begin{aligned}
\mathbf{KL}(Q||P) &= -\sum_{\mathbf{X}} Q(\mathbf{X}) \log \frac{Q(\mathbf{X})}{P(\mathbf{X})} \\
&= -\sum_{\mathbf{X}} Q(\mathbf{X}) \log P(\mathbf{X}) + \sum_{\mathbf{X}} Q(\mathbf{X}) \log Q(\mathbf{X}) \\
&= -\mathbf{E}_{\mathbf{X}\sim Q}[\log P(\mathbf{X})] + \mathbf{E}_{\mathbf{X}\sim Q}[\log Q(\mathbf{X})] \\
&= \mathbf{E}_{\mathbf{X}\sim Q}[\mathbf{E}(\mathbf{X})] + \mathbf{E}_{\mathbf{X}\sim Q}[\log Z] + \sum_i \mathbf{E}_{X_i\sim Q_i}[\log Q_i(X_i)] \\
&= \mathbf{E}_{\mathbf{X}\sim Q}[\mathbf{E}(\mathbf{X})] + \log Z + \sum_i \mathbf{E}_{X_i\sim Q_i}[\log Q_i(X_i)]
\end{aligned}
\tag{A.4}
$$

The mean-field approximation inference [109] attempts to minimize this KL-divergence, as shown in equation A.4. The approximate marginal distribution $Q_i(X_i)$ that minimizes the KL-divergence in equation A.4 is found by considering the fixed-point equations that must hold at the stationary points. Koller *et al.* gave the proof and detailed derivation in chapter 11.5 of [109]. This leads to the update equation for $Q_i(x_i)$ shown as follows.

$$
\begin{aligned}
Q_i(x_i) &= \frac{1}{Z_i} \exp\{-\psi_u(x_i) - \sum_{l'} \sum_{j\neq i\in\mathcal{N}} Q_j(x_j = l')\psi_p(x_i, x_j)\} \\
&= \frac{1}{Z_i} \exp\{-\psi_u(x_i) - \sum_{l'} \sum_{j\neq i\in\mathcal{N}} Q_j(x_j = l')[\mu(l, l') \sum_{m=1}^{K} w^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j)]\} \\
&= \frac{1}{Z_i} \exp\{-\psi_u(x_i) - \sum_{l'} \mu(l, l') \sum_{m=1}^{K} w^{(m)} \sum_{j\neq i\in\mathcal{N}} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(x_j = l')\},
\end{aligned}
\tag{A.5}
$$

where Z_i is a constant which normalises the approximate marginal distribution at pixel i. If the updates in equation A.5 are made sequentially across pixels $i = 1, ..., N$ (updating and normalising the L values $Q_i(x_i = l), l = 1, ..., L$ at each iteration), the KL-divergence is guaranteed to decrease (see the proof in chapter 11.5 of Koller *et al.* [109]). In Krähenbühl *et al.* [110], this is implemented by doing parallel updates in order to sacrifice the theoretical guarantees for speed. Although without theoretical guarantees, this parallel updates are working well empirically. Krähenbühl *et al.* [110] implemented this update as presented in Algorithm 3. One computational bottleneck is the summation in the message passing step, which is $O(N^2)$ with naive method.

A.3 Message Passing as a Convolution in High-Dimensional Space

In Krähenbühl *et al.* [110], the summation step in message passing is expressed as a convolution with a Gaussian kernel G_m in feature space. According to sampling theorem [3],

Algorithm 3 filter-based mean-field approximate inference in fully-connected conditional random fields [110].

$Q_i(l) \leftarrow \frac{1}{Z_i(\mathbf{U})} \exp\left(U_i(l)\right)$ for all i ▷ Initialization

while not converged **do**

$\tilde{Q}_i^{(m)}(l) \leftarrow \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l)$ for all m

▷ **Message Passing** from all X_j to all X_i

$\hat{Q}_i(l) \leftarrow \sum_{l' \in \mathcal{L}} \mu^{(m)}(l, l') \sum_m w^{(m)} \tilde{Q}_i^m(l')$

▷ **Compatibility Transform**

$\breve{Q}_i(l) \leftarrow \exp\{-\mu_u(l) - \hat{Q}_i(l)\}$

▷ **Local update**

$Q_i \leftarrow \frac{1}{Z_i(Q(\mathbf{X}))} \breve{Q}_i(l)$

▷ **Softmax Normalisation**

end while

this function can be reconstructed from a set of samples. This leads us to the following equation.

$$\begin{aligned} \tilde{Q}_i^m(x_i = l) &= \sum_{j \neq i \in \mathcal{N}} k^m(\mathbf{f}_i, \mathbf{f}_j) Q_j(x_j = l) \\ &= [G_m \otimes Q(l)](\mathbf{f}_i) - Q_i(x_i = l), \end{aligned} \tag{A.6}$$

where G_m is a Gaussian kernel corresponded to the m_{th} term of the sum, and $\otimes$ represents the convolution operation. It is possible to make this computationally efficient by using a data structure called the permutohedral lattice [3]. Using this method, the time complexity of performing approximate Gaussian convolution becomes $O(N)$, N being the number of pixels. In practice, this is implemented by performing these steps on the $Q(l)$. First, we perform the convolution by down-sampling $Q(l)$, convolving the samples with G_m, and up-sampling the results back.

Appendix B

Convolution and Deconvolution in Convolutional Neural Networks

The devil is in the detail.

Idiom

B.1 Introduction

For completeness, we include in this appendix a brief summary of the deconvolutional networks [235] or the *Fully-Convolutional Neural Networks* (FCNs) [143] for learning our unary model for semantic image segmentation. We first describe the feed-forward *Convolutional Neural Network* [78, 125, 98, 218]. Then we present the two fundamental computational blocks or layers in convolutional neural network: convolution and deconvolution (a.k.a. convolutional transpose).

B.2 Feed-forward convolutional neural network

a *Convolutional Neural Network* (CNN) can be formulated as a function f mapping data $\mathbf{x}$ to an output vector $\mathbf{y}$. In this thesis, we mainly consider image as data. In typical deep learning libraries such as Caffe [98] and MatConvNet [218], this function is implemented with *computational blocks* or *layers*, let denote this by expression $f = f_L \circ \cdots \circ f_1$. The outputs of each layer in the network are represented as $\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_L$, while the network input is denoted as $\mathbf{x}_0 = \mathbf{x}$. Each output $\mathbf{x}_l = f_l(\mathbf{x}_{l-1}; \mathbf{w}_l)$ is computed from the previous output $\mathbf{x}_{l-1}$ by applying the function f_l with parameters $\mathbf{w}_l$. The data flowing through the network has spatial structure, namely, a 3D array is denoted as $\mathbf{x}_l \in \mathbb{R}^{H_l \times W_l \times D_l}$. The first two dimensions of this 3D array are interpreted as spatial coordinates. A fourth non-singleton dimension in this array allows processing batches of images in parallel. This

is important for computational efficiency. The network is called *convolutional* because the function f_l acts as local and translation invariant operation (*i. e.* non-linear filters) [78, 218].

CNNs are used as classifiers [125] or regressors [77]. A typical example for CNNs is a classifier for ImageNet image classification. The output $\hat{\mathbf{y}} = f(\mathbf{x})$ is a vector of probabilities, for each of a $1,000$ possible image labels (*i. e.* dog, cat,...). If $\mathbf{y}$ is the ground truth label of image $\mathbf{x}$, we can measure the CNN performance by a loss function $\ell_{\mathbf{y}}(\hat{\mathbf{y}}) \in \mathbb{R}$. The parameters of CNNs can then be *adjusted* or *learned* to minimize this loss averaged over labelled images on the dataset. Learning generally uses *stochastic gradient descent* (SGD) [125] or its variants such as ADAM [105].

The fundamental operation to *learn* a network is computing the derivative of the loss with respect to the network parameters. This is obtained by using *backpropagation* algorithm [125, 218], which is an application of the chain rule for derivatives:

$$\begin{aligned} &\frac{d}{d\mathbf{w}_l^\top}\ell_{\mathbf{y}}(f(\mathbf{x};\mathbf{w}_1,\cdot,\mathbf{w}_L)) = \\ &\frac{d[\ell_{\mathbf{y}}\circ f_L\circ\cdot f_{l+1}](\mathbf{x}_l)}{d\mathbf{x}_l^\top}\frac{df_l(\mathbf{x}_{l-1};\mathbf{w}_l)}{d\mathbf{w}_l^\top} \end{aligned} \tag{B.1}$$

Because the output of this loss function is a scalar, the intermediate derivatives and its corresponding parameter have the same dimension. For instance, $d[\ell_{\mathbf{y}}\circ f_L\circ\cdot f_{l+1}]/d\mathbf{x}_l^\top$ has $H_l\times W_l\times D_l$ components, equal to the number of elements of $\mathbf{x}_l$. In contrast, the Jacobian such as $df_l/d\mathbf{x}_{l-1}^\top$ has $H_lW_lD_lH_{l-1}W_{l-1}D_{l-1}$ components.

Convolution is to compute the convolution of the input map $\mathbf{x}$ with a bank of K-dimensional filters f and biases b. Here, $\mathbf{x}\in\mathbb{R}^{H\times W\times D}$, $\mathbf{f}\in\mathbb{R}^{H^{'}\times W^{'}\times D\times D^{''}}$, $\mathbf{y}\in\mathbb{R}^{H^{''}\times W^{''}\times D^{''}}$. Formally, the output is given

$$y_{i''j''d''} = b_{d''} + \Sigma_{i'=1}^{H^{'}}\Sigma_{j'=1}^{W^{'}}\Sigma_{d'=1}^{D} f_{i'j'd}\times x_{i''+i'-1,j''+j'-1,d',d''}. \tag{B.2}$$

It is also possible to specify top-bottom-left-right padding $(P_h^-, P_h^+, P_w^-, P_w^+)$ of the input array and sub-sampling strides (S_h, S_w) of the output array

$$y_{i''j''d''} = b_{d''} + \Sigma_{i'=1}^{H^{'}}\Sigma_{j'=1}^{W^{'}}\Sigma_{d'=1}^{D} f_{i'j'd}\times x_{S_h(i''-1)+i'-P_h^-,S_w(j''-1)+j'-P_w^-,d',d''}. \tag{B.3}$$

In this expression, the array $\mathbf{x}$ is implicitly extended with zeros as needed. The size of the output is computed by

$$H^{''} = 1 + [\frac{H - H^{'} + P_h^- + P_h^+}{S_h}] \tag{B.4}$$

The input must be padded to have the same size of the filters, such that $H + P_h^- + P_h^+ \geq H^{'}$.

Convolution transpose (deconvolution) is the transpose of the convolution [218]. Let $\mathbf{x} \in \mathbb{R}^{H \times W \times D}$, $\mathbf{f} \in \mathbb{R}^{H^{'} \times W^{'} \times D \times D^{''}}$, $\mathbf{y} \in \mathbb{R}^{H^{''} \times W^{''} \times D^{''}}$ be the input tensor, filters, and output tensors, respectively. Convolution transpose is to use the filter bank $\mathbf{f}$ to convolve the output $\mathbf{y}$ to obtain the input $\mathbf{x}$. Because the convolution is a linear operation, this operation can be expressed as a matrix M such that $x = My$. For convolution transpose, this is expressed as $y = M^{\top}x$.

There are two important applications of convolution transpose. The first one is called deconvolutional networks [155], and the second one is a type of network such as a convolution decoder [9] that uses the transpose of a convolution. The second one is sometimes also implemented as data interpolation [34]. Since the convolution block supports input padding and output down-sampling [143], the convolution transpose block supports input up-sampling and output cropping [143].

Convolution transpose has a closed form solution [218]:

$$y_{i''j''d''} = \Sigma_{d'=1}^{D} \Sigma_{i'=0}^{q(H^{'},S_h)} \Sigma_{j'=0}^{q(W^{'},S_w)} f_{1+S_h i' + m(i''+P_h^{-},S_h),1+S_w j' + m(j''+P_w^{-},S_w),d'',d'} \times x_{1-i'+q(i''-P_h^{-},S_h),1-j'+q(j''+P_w^{-},S_w),d'}. \tag{B.5}$$

where $m(k,S) = (k-1) \mod S, q(k,S) = \lfloor \frac{k-1}{S} \rfloor$. (S_h, S_w) are the vertical and horizontal input up-sampling factors, $(P_h^{-}, P_h^{+}, P_w^{-}, P_w^{+})$ the output crops, and $\mathbf{x}$ and $\mathbf{f}$ are padded with zero values as needed.

The height of the output array $\mathbf{y}$ is given

$$H^{''} = S_h(H-1) + H^{'} - P_h^{-} - P_h^{+}. \tag{B.6}$$

Appendix C

Recurrent Neural Networks

Time moves in one direction, memory in another.

William Gibson

C.1 Introduction

We include a summary of the recurrent neural networks [182] in this appendix. Recurrent Neural Networks (RNN) [84, 85] are different from feed-forward convolutional neural networks. In the internal state of the network, there are recurrent connections that allow memories of previous inputs to persist, which influence the output of the network. Compared with CNNs, this unique mechanism helps RNNs to better exploiting the long-range dependencies in the data [86].

In typical RNNs, the function for a hidden layer is an element-wise version of sigmoid function. RNNs have the vanishing gradient problem when compute the very early input. This is due to the drawbacks of the RNNs' architectures [94]. In order to address this problem, Long Short-Term Memory (LSTM) [94] uses *memory cells* to store information. In addition, LSTM allows to disable writing to a cell by switching off the gate, which prevent the changes to the cell contents over iterations. when the gate is switched on again, LSTM update the cells by computing a weighted average of a new input value and the previous one. A simple RNN and a long short term memory (LSTM) are presented as follows.

C.2 Recurrent Neural Networks

Let $\mathbf{x} = (x_1, \cdots, x_T)$ be an input vector sequence, it pass through weighted connections to a stack of N hidden layers that are recurrently connected. Through this pass, we compute

first the hidden vector sequence $\mathbf{h}^n = (h_1^n, \cdots, h_T^n)$ and then the output vector sequence $\mathbf{y} = (y_1, \cdots, y_T)$. Each output vector y_t parametrize a predictive distribution $\mathbf{Pr}(x_{t+1}|y_t)$ over the possible next inputs x_{t+1}. The first element x_1 of every input sequence is a null vector whose entries are zero. For a prediction of x_2, the first actual input, there is no prior information.

In RNN, skip connections from the inputs to hidden layers make it simpler to train the networks by reducing the number of processing steps between the bottom of the network and the top, and help to address the problem of vanishing gradients [93].

The hidden layer activations are computed by iterating the following equations from $t = 1$ to T and from $n = 2$ to N:

$$h_t^1 = \mathcal{H}(W_{ih^1}x_t + W_{h^1h^1}h_{t-1}^1 + b_h^1) \tag{C.1}$$

$$h_t^n = \mathcal{H}(W_{ih^n}x_t + W_{h^{n-1}h^n}h_{t-1}^{n-1} + b_h^n) \tag{C.2}$$

where W_ih^n is the weight matrix connecting the inputs to the n_{th} hidden layer, $W_{h^1h^1}$ is the recurrent connection at the first hidden layer, the b terms denote bias vectors, and $\mathcal{H}$ is the hidden layer function.

Given the hidden layers, the output sequence is computed as follows:

$$\hat{y}_t = b_y + \sum_{n=1}^{N} W_{h^n y}h_t^n \tag{C.3}$$

$$y_t = \mathcal{Y}(\hat{y}_t) \tag{C.4}$$

where $\mathcal{Y}$ is the function for output layer.

This network gives the probability to the input sequence $\mathbf{x}$:

$$\mathbf{Pr}(\mathbf{x}) = \sigma_{t=1}^{T}\mathbf{Pr}(x_{t+1}|y_t) \tag{C.5}$$

and the sequence loss $\mathcal{L}(\mathbf{x})$ is the negative logarithm of $\mathbf{Pr}(\mathbf{x})$:

$$\mathcal{L}(\mathbf{x}) = -\sum_{t=1}^{T} \log \mathbf{Pr}(x_{t+1}|y_t) \tag{C.6}$$

The partially derivatives of the loss with respect to the weights of the network can be efficiently computed using backpropagation through time [227] applied to the computation graph.

C.3 Long Short-Term Memory

LSTM architecture [85] uses $\mathcal{H}$ that is defined as follows.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{C.7}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{C.8}$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{C.9}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{C.10}$$

$$h_t = o_t \tanh(c_t) \tag{C.11}$$

where σ, i, f, o and c are respectively the logistic sigmoid function, the *input gate*, *forget gate*, *output gate*, *cell* and *cell input* activation vectors. All of these vectors are the same size as the hidden vector h. W_{xi}, W_{hi}, W_{ci}, W_{xf}, W_{hf}, W_{cf}, W_{xc}, W_{hc}, W_{xo}, W_{ho}, W_{co} are respectively the input-input gate weight matrix, the hidden-input gate weight matrix, the weight matrix from cell input to input gate, the input-forget gate weight matrix, the hidden-output gate weight matrix, the cell-forget gate weight matrix, the weights input-cell,hidden-cell, the weight matrix for the input-output gate, the hidden-output gate weight matrix , the cell-output matrix. The bias terms are omitted from equations above for the clarity.

Appendix D

Bibliography on semantic segmentation

> If I have seen further than others, it is by standing upon the shoulders of giants.
>
> *Isaac Newton*

This appendix describes some of exciting research related to this thesis. Although not all papers are directly categorized, we have arranged this review hierarchically as far as possible. We first summarize the literature in the relevant research from the image re-organization [144] area. Then we present the research in semantic image segmentation before deep learning era and after deep learning era. We detailed discuss the research related to the fully convolutional neural networks [143] and CRF as RNN [241] (described in chapter 4).

D.1 From segmentation to semantic image segmentation

We briefly summarize the research in image re-organization.

Image segmentation is the partitioning of an image into multiple sets of pixels. Image segmentation, also known as image re-organization [144], provides the foundation for higher-level computer vision tasks [146]. This is also known as super-pixels segmentation or unsupervised segmentation. There is a large literature on segmentation, dating back over 40 years, with applications in many areas including computer vision. The representative approach such as normalize-cut [188] tries to group pixels based on their similarity. These generated super-pixels should align well with the boundaries of objects. However, this might not hold in practice due to the cluttered background and the faint objects edges. Comaniciu and Meer [49] developed an effective image segmentation approach based on

Mean-shift clustering. Felzenszwalb and Huttenlocher [75] employed a graph-based representation to define the prediction for measuring the evidence for a boundary between two regions. Based on these predictions, they developed an efficient segmentation algorithm based on this prediction. Arbelaez *et al.* [6] developed a benchmark BSD500 and BSD300 for evaluating the algorithms for image segmentation and boundary detection.

Foreground segmentation is to extract foreground object from an image. There are two settings to solve this problem. One setting is to consider this problem as a special case of semantic image segmentation, and one can address this problem with supervised learning algorithm. The other one is in an interactive setting, where users are required to provide extra information about the segmentation. This problem is strongly related to one direction in saliency region detection. In this direction of saliency region detection research, the goal is to automatically segment the salient regions. Rother et al. [178] and Lempitsky *et al.* [126] show it is possible to achieve foreground segmentation with bounding box prior. This prior can either come from the user interactive or from an object detector. Foreground segmentation is strongly related to the direction of salient region detection. Achanta *et al.* [1] presented a method to determine salient regions in images using low-level features of luminance and colour. Liu *et al.* [141] developed a system that combines the hand-craft features and conditional random field for segmenting the foreground objects from images, and they defined this as salient region detection and segmentation, which is similar to the foreground segmentation research. In Computer Graphics, Image Matting [167] is referred to the problem of accurate foreground estimation in images and video. Porter and Duff [167] established the mathematical formulation for this problem. Specifically, they introduced alpha channel as the means to control the linear interpolation of foreground and background colours for anti-aliasing purposes when rendering a foreground over an arbitrary background. Rhemann *et al.* [173] developed a benchmark for evaluating the image matting algorithms.

Co-segmentation is referred to segment the common objects from multiple images. Different from foreground segmentation or interactive segmentation, co-segmentation is to exploit the the weakly supervision information from the availability of multiple images that contain instances of the same objects. Rother *et al.* [180] first introduces the idea of image co-segmentation in a setting where the same objects are in front of different backgrounds in a pair of images. There are several works along this line. The methods in [99, 221] addressed co-segmentation without explicityly encode the ”objectness” assumption. Vicente *et al.* [222] proposed a solution that works with a pool of proposal segmentation for object co-segmentation. Joulin *et al.* [100] developed an energy-minimization approach that could handle multiple classes and larger number of images. This approach combines

spectral- and discriminative- clustering terms, is optimized using EM method, and is initialized using a convex quadratic approximation of the energy. Rubio *et al.* [181] generalize the idea to video co-segmentation. Given a video sequence that contains the same object (or objects belonging to the same category) moving in a similar manner, it aims to outline the regions in all frames.

Instance segmentation is trying to assign each pixel with an object instance label. This is considered to be an upgrade version from object detection. In generic object detection, the correct prediction is required to have at least 0.5 intersection-over-union overlap. While this requirement does not satisfy applications like autonomous vehicles, in which the correctly detections should have at least 0.7 intersection-over-union overlap. In contrast to foreground segmentation and semantic image segmentation, instance segmentation requires to not only find out the object class label, but also seeks for distinguishing different instances of the same object class. This is a difficult problem, there are many ongoing researches about this problem by the time of writing this thesis. Hariharan *et al.* [88] developed a solution that simultaneously detect and segment the objects. It tailed R-CNN [81] for instance segmentation task. Silberman *et al.* [194] introduced a coverage loss function that helps jointly inferring dense semantic and instance labels for indoor scenes. Dai *et al.* [54] exploited the shape information via masking convolutional features for instance segmentation. Hariharan *et al.* [89] introduced hypercolumns as pixel descriptors for semantic image segmentation, instance segmentation, and fine-grained recognition. This hypercolumn at a pixel is defined as the vector of activations of all CNN units above that pixel. Chen *et al.* [40] addressed the occlusion problems in instance segmentation by incorporating top-down category specific reasoning and shape prediction through exemplars into an energy minimization framework. In contrast to other works influenced by the ideas of region proposals [81], Liang *et al.* [133] introduced a proposal-free network that directly outputs the instance numbers of different categories. Similarly, Liu *et al.* [140] proposed a Multi-scale Patch Aggregation framework that predicts instance segmentation without generating proposals. In the direction of instance segmentation for the application of autonomous vehicles, Zhang [238] generalized CNN-CRF frameworks for instance segmentation.Dai [55] developed a multi-task Network Cascades for instance-aware semantic segmentation. Li and Malik [129] introduced an iterative approach for instance segmentation. Liang *et al.* [132] developed a complex networks that recursively predict the instance segmentation. It consists of a reversible proposal refinement sub-network that predicts bounding box offsets to refine the location of object proposals, and an instance-level segmentation sub-network that generates foreground mask of the dominant object instance

in each proposal. Zhang *et al.* [237] formulated the global labeling problem with fully-connected CRF [110] and improved CNN-CRF framework [238] for instance segmentation. Romera-Paredes [174] investigate the use of fully convolutional neural networks and long-short term memory networks in instance segmentation.

Semantic Image Segmentation prefers to assign a class label to each pixel in the image. This problem is important to holistic scene understanding. It combines two Computer Vision problems: recognition and reorganization. A lot of research solutions have been developed to tackle this problem over times. We summarize exciting researches in next section.

D.2 Semantic Image Segmentation before Deep Learning

The works in semantic image segmentation can be traced back to Duygulu *et al.* [62]. Shotton *et al.* [192] developed a semantic image segmentation based on TextonBoost. The features used in TextonBoost are texton, location, colour features. Texton is the clusters of filter-bank responses. TextonBoost uses joint boosting trained on these features, in which different classes share features and weak classifier is based on counting features. Shotton *et al.* [191] further optimize the speed of their system by making use of the Decision Forest instead of joint boost. One notable problems within TextonBoost is that the unary predictions are often noisy. In TextonBoost, GraphCut-based CRFs approach has been employed to solve this problem. However, due to the limitation of pair-wise CRFs, the longer connectivities in images are not captured. Kohli *et al.* [107] and Ladicky *et al.* [116] have developed higher-order CRFs approaches to solve this problem. By making use of the efficient filtering approach such as permutohedral lattice, Krähenbühl *et al.* [110].

Alternative way of solving semantic image segmentation is to use non-parametric approaches. This starts by finding the pixel-wise correspondence using approaches like SIFT Flow [137]. SIFT flows treat the semantic image segmentation in a different way, by aligning an image to its nearest neighbors in a large image corpus containing a variety of scenes. Tighe and Lazebnik [208] has proposed a scalable non-parametric parsing system. In this system, one first performs global scene-level matching against the training set, followed by super-pixel-level matching and post-processing with Markov Random Fields (MRFs) to incoprporate neighborhood context. Exemplar SVM is another example in this line of research, Tighe and Lazebnik [207] developed a parsing system that combines per-exemplar detector and region-based parsing.

Ladicky *et al.* [116, 115] have used super-pixel as a higher order potential in CRFs. Different from this, Carreira *et al.* [30] developed a semantic segmentation system that

makes use of object region proposals. It first generate diverse foreground object region proposals through parametric min-cuts. It then achieve semantic image segmentation by ranking the object region proposals by classifying them. This semantic image segmentation can also be further refined by using Markov Random Fields as post-processing. Batra*et al.* [12] and Yadollahpour *et al.* [163] developed diverse M-Best algorithm. Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

D.3 Deep learning for semantic image segmentation and low-level computer vision problems

Deep learning approaches [123] including Convolutional neural networks (CNN) [125] and recurrent neural networks (RNN) [86, 199] have recently dramatically improved the state-of-the-art in object recognition. This type of approach [123] allows computational models that are composed of multiple processing layers to learn data representation with multiple levels of abstraction. By making use of back-propagation algorithm on GPUs, modern deep learning algorithms efficiently learn the parameters for the computational model from large data. Semantic image segmentation and other related low-level computer vision problems such as instance segmentation, image denoising, stereo matching and optical flow are considered to be structured output prediction problems. One question inspired by the success of deep learning is how to leverage the deep learning approaches for structured output prediction problems like semantic image segmentation and other low-level computer vision tasks. In this section, we review the development history of applying deep learning for semantic image segmentation and other low-level computer vision problems.

In ImageNet [58] image classification competition 2012, Krizhevsky *et al.* [113] showed a Convolutional neural network (CNN) implemented in GPUs perform significant better than traditional approaches. This started the work of applying Convolutional neural networks in many computer vision tasks as well as other artificial intelligent tasks. Farabet *et al.* [69] trained a multi-scale CNN first time on semantic image segmentation task.

However, he did not explore the fine-tuning ideas, the performance improvement was not very significant. Girshick *et al.* [79] first show the CNN classification model trained on ImageNet could be generalised to object detection on Pascal VOC dataset [68]. This way of fine-tuning a ImageNet classification model on objet detection leads to dramatically higher object detection performance. This success inspired many works for applying CNN for semantic image segmentation through fine-tuning ImageNet classification models. In particularly, Long *et al.* [143] have shown significant accuracy boost for semantic image segmentation by fine-tuning the VGG image classification models. They proposed a deconvolutional layer which effectively upsample the resolution of feature maps to that of the original input image. This deconvolutional layer can be implemented as transpose of convolutional operation [218], which is a common use operation in CNN. The key insight among these works is to learn strong feature representation and classifiers in an end-to-end system instead of hand-crafting features with heuristic parameter tuning. This key insight has motivated a wide variety of approaches for semantic image segmentation using deep learning. These approaches can be categorized into two directions.

The first direction is based on the idea of marrying bottom-up semantic image segmentation [5] with deep learning. This is to utilize separate mechanisms for feature extraction, and image segmentation. The representative work [151] along this direction is the application of a CNN for the extraction of meaningful features, and using super-pixels to account for the structural pattern of the images. Other representative example [69] attempted to first obtained super-pixels from images and then used a feature extraction process on each of them. The main disadvantages of this direction of approaches is that errors in the initial proposals may lead to poor predictions, regardless of how well the feature extraction process. Bell *et al.* [13] proposed sliding-window-based CNN for segmenting material from images. Cimpoi *et al.* [48] built on top of R-CNN [79] pipeline for segmenting the material from images. On top of the CNN, these works also apply Dense CRF [110] as a post-processing step to further improve the consistency of segmentation. In contrast to these works, Pinherio and Collobert [166] proposed a Recurrent Neural Network (RNN) to model the spatial dependencies during scene parsing.

The second direction is to directly learn a nonlinear model from the images to the label map. Eigen *et al.* [66] replaced the last fully connected layers of a classification CNN by convolutional layers to keep spatial information. They showed impressive results for predicting depth from single images. Long *et al.* [143] used the concept of fully convolutional networks, and the notion that top layers of CNN obtain meaningful features for object recognition whereas low layers in CNN keep the information about the structure of image such as edges. They showed that a deconvolutional layer can be integrated

into CNN to achieve end-to-end pixel-wise labeling results. Ronnerger *et al.* [175] and Noh *et al.* [155] have also shown variants architecture for pixel-wise labeling based on similar ideas around deconvolutional layer. The simplest version of this deconvolution can be implemented as convolution transpose [218]. Along this line, Chen *et al.* [34], Liu *et al.* [138] and Yu *et al.* [234] further improved the approach of Long *et al.* [143] with different architectures.

CNN-based approaches alone these two directions showed very significant accuracy boost for semantic image segmentation task, compared against its traditional approach counterparts. However, the upsampled results are often noisy and the boundaries of the objects are missing. In order to address these problems, a series of works appeared to combine the deep Convolutional Neural Networks and Markov Random Fields. Bell *et al.* [13] and Chen *et al.* [34] used a CRF to refine segmentation results obtained from a CNN. But this post-processing steps break the story of end-to-end training CNN and achieve suboptimal results on Pascal VOC dataset. Zheng *et al.* [241] showed an end-to-end trainable approach to integrate both CNN and Dense CRF. Schwing *et al.* [184] showed concepts proofs about a similar idea. Liu [138] showed further improvements could be achieved by integrating extra CNNs into CRF framework. Lin [135] developed a complicated CNN-CRF-based system which have an extra CNN for generating pairwise potentials. Arnab *et al.* [7] incorporated higher-order potential functions in CRF-RNN, and achieved the top results by the time it was published in Pascal VOC. Arnab *et al.* [8] further generalized this framework to work with instance segmentation problem.

Works that use deep learning for structured output predictions are also found in different domains. For example, Do *et al.* [59] proposed to combine deep neural networks and markov networks for sequence labeling tasks. Jain *et al.* [97] presented a CNN can perform well like MRFs/CRFs approaches in image restoration application. Bottou [23] showed the benefits of combination of CNNs and structured loss in document recognition. Peng *et al.* [164] used a modified version of CRFs for the same purpose. Related to this line of works, Jaderberg *et al.* [96] showed a CNN-CRF model for text recognition on natural images. Tompson *et al.* [210] demonstrated a joint CNN and CRF model could be used for human pose estimation. Chen *et al.* [35] focused on image classification task with a similar approach. Girshick *et al.* [80] express deformable part models, a special MRF model, as a layer in a neural network.

D.4 Related to Fully Convolutional Networks

Fully Convolutional Network is effective in pixel-wise labelling tasks including semantic image segmentation, dense correspondence estimation, and etc.

LeCun *et al.* [124] pioneered the first application for digits recognition using convolutional neural network and back-propagation algorithm. Matan *et al.* [147] extended convolutional neural networks to work with arbitrary-size input. Matan *et al.* [147] applied Viterbi decoding to obtain their outputs. Wolf and Platt [229] expanded the outputs of convolutional neural networks to 2-dimensional maps of detection. Both of these works involve fully convolutional inference and learning for detection.

Several recent works also explored the use of convolutional neural networks in dense prediction. In the application of localizing cells and nuclei in microscopic images, Ning *et al.* [154] developed a coarse multi-class segmentation system based on a convolutional neural network with fully convolutional inference. Farabet *et al.* [69] developed multi-scale convolutional neural networks and conditional random fields for semantic image segmentation. Sermanent *et al.* [185] applied convolutional neural networks with fully convolutional inference in the sliding window detection. Pinheiro and Collobert [166] explore the use of recurrent neural networks and fully convolutional inference in semantic image segmentation. Eigen *et al.* [65, 64, 66] investigated image restoration and depth estimation using fully convolutional inference. Fully convolutional training is rare, but used effectively by Tompson *et al.* [210] to learn an end-to-end part detector and spatial model for pose estimation.

Long *et al.* [143] developed fully convolutional neural networks for semantic image segmentation. Their work has achieved the state-of-the-art performance by the time it was published. It has also integrated into a popular deep learning and computer vision library Caffe [98]. DeepLab models [34] raise output resolution by dilated (A.K.A. Atrous) and dense CRF [110] inference. Three works including Chen *et al.* [34], Bell *et al.* [13] and Cimpoi *et al.* [48] investigated the two-stage CNN-CRF pipeline. Bell [13] and Cimpoi *et al.* [48] focus on material segmentation, while Chen *et al.* [34] focus on semantic image segmentation. Joint CRFasRNN model [241] is an end-to-end integration of the CRF for further improvement. Independent work also appears in Schwing *et al.* [184]. ParseNet [142] normalizes the features for fusion and captures context with global pooling. The deconvolutional network [155] approach restores resolution by proposals, stacks of learned deconvolution, and unpooling. U-Net [175] combines skip layers and learned deconvolution for pixel labeling of microscopy images. Similar to Atrous [34], the dilation [234] architecture makes through use of dilated convolution for pixel-precise out-

put without a random field or skip layers. Dai *et al.* [53] developed a weakly supervised method for semantic image segmentation based on iterating between automatically generating region proposals and training convolutional networks. Pathak *et al.* [161] proposed a novel multiple-instance learning formulation of multi-class semantic segmentation learning by a fully convolutional network. Bertasius [17] generalized multi-scale fully-convolutional networks for contour detection. Fischer *et al.* [77] investigated the use of fully convolutional neural network in optical flow estimation. Thewlis *et al.* [205] developed a fully trainable deep matching network, which formulates the deep matching [172] as a U-net [175]. Pfister *et al.* [165] investigated a convolution neural network architecture that is able to benefit from temporal context by combining information across the multiple frames using optical flow. Xie and Tu [232] developed an edge detection system based on multi-scale fully-convolutional networks. Dai *et al.* [56] developed a region-based object detection using fully convolutional networks and region proposal networks [171]. Dai *et al.* [52] generalized the fully convolutional networks for instance segmentation.

Bibliography

[1] Radhakrishna Achanta, Francisco Estrada, Patricia Wils, and Sabine Süsstrunk. Salient region detection and segmentation. In *International Conference on Computer Vision Systems*, 2008.

[2] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Süsstrunk. Frequency-tuned salient region detection. In *IEEE CVPR*, pages 1597–1604, 2009.

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

[4] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. In *IEEE CVPR*, 2012.

[5] Pablo Arbeláez, Bharath Hariharan, Chunhui Gu, Saurabh Gupta, Lubomir Bourdev, and Jitendra Malik. Semantic segmentation using regions and parts. In *IEEE CVPR*, 2012.

[6] Pablo Arbeláez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, 2011.

[7] Anurag Arnab, Sadeep Jayasumana, Shuai Zheng, and Philip Torr. Higher order conditional random fields in deep neural networks. In *ECCV*, 2016.

[8] Anurag Arnab and Philip Torr. Bottom-up instance segmentation using deep higher-order crfs. In *BMVC*, 2016.

[9] Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. In *CoRR abs/1505.07293*, 2015.

[10] Adrian Barbu. Training an active random field for real-time image denoising. *IEEE TIP*, 18(11):2451–2462, 2009.

[11] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24:1–11, 2009.

[12] Dhruv Batra, Payman Yadollahpour, Abner Guzman-Rivera, and Gregory Shakhnarovich. Diverse m-best solutions in markov random fields. In *ECCV*, 2012.

[13] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. In *IEEE CVPR*, 2015.

[14] Yoshua Bengio, Yann LeCun, and Donnie Henderson. Globally trained handwritten word recognizer using spatial representation, convolutional neural networks, and hidden markov models. In *NIPS*, pages 937–937, 1994.

[15] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 1994.

[16] Brent Berlin and Paul Kay. *Basic color terms: Their universality and evolution*. Univ of California Press, 1991.

[17] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *IEEE CVPR*, 2015.

[18] Floraine Berthouzoz, Wilmot Li, Mira Dontcheva, and Maneesh Agrawala. A framework for content-adaptive photo manipulation macros: Application to face, landscape, and global manipulations. *ACM Trans. Graph.*, 30(5):120, 2011.

[19] Christopher M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[20] Andrew Blake, Pushmeet Kohli, and Carsten Rother. *Markov random fields for vision and image processing*. Mit Press, 2011.

[21] Andrew Blake, Carsten Rother, Matthew Brown, Patrick Perez, and Philip Torr. Interactive image segmentation using an adaptive gmmrf model. In *ECCV*, pages 428–441, 2004.

[22] Richard A Bolt. Put-that-there: Voice and gesture at the graphics interface. In *ACM SIGGRAPH*, pages 262–270, 1980.

[23] Leon Bottou, Yoshua Bengio, and Yann Le Cun. Global training of document processing systems using graph transformer networks. In *IEEE CVPR*, 1997.

[24] Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *IEEE ICCV*, pages 105–112, 2001.

[25] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, 2004.

[26] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *IEEE ICCV*, volume 1, pages 105–112, 2001.

[27] Gary Bradski et al. The opencv library. *Doctor Dobbs Journal*, 25(11):120–126, 2000.

[28] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[29] Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie. Visual recognition with humans in the loop. In *ECCV*, pages 438–451, 2010.

[30] Joao Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. Free-form region description with second-order pooling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014.

[31] Robert Carroll, Aseem Agarwala, and Maneesh Agrawala. Image warps for artistic perspective manipulation. *ACM Trans. Graph.*, 29(4):127, 2010.

[32] Yuning Chai, Victor Lempitsky, and Andrew Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *IEEE ICCV*, 2013.

[33] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.

[34] 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.

[35] Liang-Chieh Chen, Alexander G. Schwing, Alan L. Yuille, and Raquel Urtasun. Learning deep structured models. In *ICLRW*, 2015.

[36] Qifeng chen and Vladlen Koltun. Fast mrf optimization with application to depth reconstruction. In *IEEE CVPR*, 2014.

[37] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2photo: Internet image montage. *ACM Trans. Graph.*, 28(5):124:1–10, 2009.

[38] Tao Chen, Ping Tan, Li-Qian Ma, Ming-Ming Cheng, Ariel Shamir, and Shi-Min Hu. Poseshop: Human image database construction and personalized content synthesis. *Visualization and Computer Graphics, IEEE Transactions on*, 19(5):824–837, 2013.

[39] Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2net: Accelerating learning via knowledge transfer. In *ICLR*, 2016.

[40] Yi-Ting Chen, Xiaokai Liu, and Yang Ming-Hsuan. Multi-instance object segmentation with occlusion handling. In *IEEE CVPR*, 2015.

[41] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. Salient object detection and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014.

[42] Ming-Ming Cheng, NiloyJ. Mitra, Xiaolei Huang, and Shi-Min Hu. Salientshape: group saliency in image collections. *The Visual Computer*, 30(4):443–453, 2014.

[43] Ming-Ming Cheng, Victor Adrian Prisacariu, Shuai Zheng, Philip H.S. Torr, and Carsten Rother. Densecut: densely connected CRFs for realtime GrabCut. *Computer Graphics Forum*, 34(7), 2015.

[44] Ming-Ming Cheng, Jonathan Warrell, Wen-Yan Lin, Shuai Zheng, Vibhav Vineet, and Nigel Crook. Efficient salient region detection with soft image abstraction. In *IEEE ICCV*, 2013.

[45] Ming-Ming Cheng, Fang-Lue Zhang, Niloy J. Mitra, Xiaolei Huang, and Shi-Min Hu. RepFinder: Finding Approximately Repeated Scene Elements for Image Editing. *ACM Trans. Graph.*, 29(4):83:1–8, 2010.

[46] Ming-Ming Cheng, Guo-Xin Zhang, Niloy J. Mitra, Xiaolei Huang, and Shi-Min Hu. Global contrast based salient region detection. In *IEEE CVPR*, pages 409–416, 2011.

[47] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip H. S. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *IEEE CVPR*, 2014.

[48] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, and Andrea Vedaldi. Deep filter banks for texture recognition, description, and segmentation. *IJCV*, 30(1):1–30, 2016.

[49] 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.

[50] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE CVPR*, 2016.

[51] Harald Cramér. *Mathematical Methods of Statistics*. Princeton University Press, 1946.

[52] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In *arXiv:1603.08678*, 2016.

[53] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *IEEE ICCV*, 2015.

[54] Jifeng Dai, Kaiming He, and Jian Sun. Convolutional feature masking for joint object and stuff segmentation. In *IEEE CVPR*, 2015.

[55] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *IEEE CVPR*, 2016.

[56] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *arXiv:1605.06409*, 2016.

[57] Thomas Dean, Mark A. Ruzon, Mark Segal, Jonathon Shlens, Sudheendra Vijayanarasimhan, and Jay Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *IEEE CVPR*, 2013.

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

[59] Trinh-Minh-Tri Do and Thierry Artieres. Neural conditional random fields. In *NIPS*, 2010.

[60] Justin Domke. Learning graphical model parameters with approximate marginal inference. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(10):2454–2467, 2013.

[61] Jian Dong, Qiang Chen, Shuicheng Yan, and Alan Yuille. Towards unified object detection and semantic segmentation. In *ECCV*, 2014.

[62] Pinar Duygulu, Kobus Barnard, Joao F. G. de Freitas, and David A. Forsyth. Object recognition as machine translation:learning a lexicon for a fixed image vocabulary. In *ECCV*, 2002.

[63] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. *ACM Trans. Graph.*, pages 341–346, 2001.

[64] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *arXiv:1411.4734*, 2014.

[65] David Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *IEEE ICCV*, 2013.

[66] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014.

[67] Mark Everingham, S. M. Ali Eslami, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015.

[68] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[69] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013.

[70] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *IEEE CVPR*, pages 1–8, 2009.

[71] Ali Farhadi, Ian Endres, and Derek Hoiem. Attribute-centric recognition for cross-category generalization. In *IEEE CVPR*, 2010.

[72] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *IEEE CVPR*, 2009.

[73] Ali Farhadi and Mohammad Amin Sadeghi. Recognition using visual phrases. In *IEEE CVPR*, pages 1745–1752, 2011.

[74] Pedro Felzenszwalb, Ross Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, 2010.

[75] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004.

[76] Vittorio Ferrari and Andrew Zisserman. Learning visual attributes. *NIPS*, 2007.

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

[78] Kunihiko Fukushima and Sei Miyake. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, 15(6):455–469, 1982.

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

[80] Ross Girshick, Forrest Iandola, Trevor Darrell, and Jitendra Malik. Deformable part models are convolutional neural networks. In *IEEE CVPR*, 2015.

[81] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(1):142–158, 2016.

[82] Chen Goldberg, Tao Chen, Fang-Lue Zhang, Ariel Shamir, and Shi-Min Hu. Data-driven object manipulation in images. *Comput. Graph. Forum*, 31:265–274, 2012.

[83] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *IEEE ICCV*, 2009.

[84] Alex Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012.

[85] Alex Graves. Generating sequences with recurrent neural networks. In *arXiv:1308.0850*, 2014.

[86] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jűrgen Schmidhuber. A novel connectionist system for improved unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5), 2009.

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

[88] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *ECCV*, 2014.

[89] Bharath Hariharan, Pablo Arbelaez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *IEEE CVPR*, 2015.

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

[91] Geremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. In *ECCV*, 2008.

[92] Steven Henderson. Augmented Reality for Maintenance and Repair. "http://www.youtube.com/watch?v=mn-zvymlSvk, 2008.

[93] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jűrgen Schmidhuber. *Gradient flow in recurrent nets: the difficulty of learning long-term dependencies*. IEEE Press, 2001.

[94] Sepp Hochreiter and Jűrgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[95] Sunnybrook Hospital. Xbox Kinect in the hospital operating room. "http://www.youtube.com/watch?v=f5Ep3oqicVU, 2008.

[96] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep structured output learning for unconstrained text recognition. In *ICLR*, 2015.

[97] Viren Jain, Joseph F Murray, Fabian Roth, Srinivas Turaga, Valentin Zhigulin, Kevin L Briggman, Moritz N Helmstaedter, Winfried Denk, and H Sebastian Seung. Supervised learning of image restoration with convolutional networks. In *IEEE ICCV*, 2007.

[98] 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*, pages 675–678, 2014.

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

[100] Armand Joulin, Francis R. Bach, and Jean Ponce. Multi-class cosegmentation. In *IEEE CVPR*, 2012.

[101] Fahad Shahbaz Khan, Rao Muhammad Anwer, Joost van de Weijer, Andrew D. Bagdanov, Maria Vanrell, and Antonio M. López:. Color attributes for object detection. In *IEEE CVPR*, pages 3306–3313, 2012.

[102] Martin Kiefel and Peter V. Gehler. Human pose estmation with fields of parts. In *ECCV*, 2014.

[103] B. Kim, M. Sun, P. Kohli, and Silvio Savarese. Relating things and stuff by high-order potential modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014.

[104] Junhwan Kim and Ramin Zabih. Factorial markov random fields. In *ECCV*, 2002.

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

[106] Alexander Kirillov, Dmitrij Schlesinger, Shuai Zheng, Bogdan Savchynskyy, Philip Torr, and Carsten Rother. Efficient likelihoood learning of a generic cnn-crf model for semantic segmentation. In *ACCV*, 2016.

[107] Pushmeet Kohli, Lubor Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. In *IEEE CVPR*, 2008.

[108] Pushmeet Kohli, Lubor Ladicky, and Philip H. S. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 82(3):302–324, 2009.

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

[110] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011.

[111] Philipp Krähenbühl and Vladlen Koltun. Parameter learning and convergent inference for dense random fields. In *ICML*, 2013.

[112] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations. In *IEEE CVPR*, 2016.

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

[114] Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Baby talk: Understanding and generating simple image descriptions. In *IEEE CVPR*, pages 1601–1608, 2011.

[115] Lubor Ladicky, Chris Russell, Pushmeet Kohli, and Philip HS Torr. Graph cut based inference with co-occurrence statistics. In *ECCV*, 2010.

[116] Lubor Ladicky, Christopher Russell, Pushmeet Kohli, and Philip HS Torr. Associative hierarchical crfs for object class image segmentation. In *IEEE ICCV*, 2009.

[117] Lubor Ladicky, Paul Sturgess, Karteek Alahari, Chris Russell, and P. H. S. Torr. What,where & how many? combining object detectors and crfs. In *ECCV*, 2010.

[118] Lubor Ladicky, Paul Sturgess, Christopher Russell, Sunando Sengupta, Yalin Bastanlar, William F. Clocksin, and Philip H. S. Torr. Joint optimisation for object class segmentation and dense stereo reconstruction. In *BMVC*, 2010.

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

[120] Jean-François Lalonde, Derek Hoiem, Alexei A. Efros, Carsten Rother, John M. Winn, and Antonio Criminisi. Photo clip art. *ACM Trans. Graph.*, 26(3):3, 2007.

[121] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *IEEE CVPR*, pages 951–958, 2009.

[122] Gierad Laput, Mira Dontcheva, Gregg Wilensky, Walter Chang, Aseem Agarwala, Jason Linder, and Eytan Adar. Pixeltone: A multimodal interface for image editing. In *CHI*, 2013.

[123] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[124] Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, R. E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. Handwritten digit recognition with a back-propagation network. In *NIPS*, 1989.

[125] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[126] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *IEEE ICCV*, 2009.

[127] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(2):228–242, 2008.

[128] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *IEEE CVPR*, 2016.

[129] Ke Li, Bharath Hariharan, and Jitendra Malik. Iterative instance segmentation. In *IEEE CVPR*, 2016.

[130] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM SIGGRAPH*, 23(3):303–308, 2004.

[131] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3):303–308, 2004.

[132] Xiaodan Liang, Yunchao Wei, Xiaohui Shen, Zequn Jie, Jiashi Feng, Liang Lin, and Shuicheng Yan. Reversible recursive instance-level object segmentation. In *IEEE CVPR*, 2016.

[133] Xiaodan Liang, Yunchao Wei, Xiaohui Shen, Jianchao Yang, Liang Lin, and Shuicheng Yan. Proposal-free network for instance-level object segmentation. In *arXiv preprint arXiv:1509.02636*, 2015.

[134] Guosheng Lin, Chunhua Shen, Ian Reid, and Anton van dan Hengel. Efficient piecewise training of deep structured models for semantic segmentation. In *IEEE CVPR*, 2016.

[135] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollar. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.

[136] Wen-Yan Lin, Ming-Ming Cheng, Shuai Zheng, J. Lu, and N. Crook. Robust non-parametric data fitting for correspondence modeling. In *IEEE ICCV*, 2013.

[137] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):978–994, 2011.

[138] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. In *IEEE CVPR*, 2015.

[139] J. Liu, J. Sun, and H.-Y. Shum. Paint selection. *ACM Trans. Graph.*, 28(3), 2009.

[140] Shu Liu, Xiaojuan Qi, Jianping Shi, Hong Zhang, and Jiaya Jia. Multi-scale patch aggregation (mpa) for simultaneous detection and segmentation. In *IEEE CVPR*, 2016.

[141] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, Tang X., and Shum H.Y. Learning to detect a salient object. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(2):353–367, 2011.

[142] Wei Liu, Andrew Rabinovich, and Alexander C. Berg. Parsenet: Looking wider to see better. In *arXiv:1506.04579*, 2015.

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

[144] Jitendra Malik, Pablo Andrés Arbeláez , João Carreira, Katerina Fragkiadaki, Ross B. Girshick, Georgia Gkioxari, Saurabh Gupta, Bharath Hariharan, Abhishek Kar, and Shubham Tulsiani. The three r's of computer vision: Recognition, reconstruction and reorganization. *Pattern Recognition Letters*, 2016.

[145] Tomasz Malisiewicz and Alexei A. Efros. Recognition by association via learning per-exemplar distances. In *IEEE CVPR*, June 2008.

[146] David Marr. *Vision: a computational investigation into the human representation and processing of visual information*. MIT Press, 1982.

[147] O. Matan, C. J. Burges, Y. LeCun, and J. S. Denker. Multi-digit recognition using a space displacement neural network. In *NIPS*, 1991.

[148] Moritz Menze, Christian Heipke, and Andreas Geiger. discrete optimisation optical flow estimation. In *German Conference on Pattern Recognition*, 2015.

[149] Microsoft, January 2012. http://www.micro soft.com/ download/details.aspx?id=27226.

[150] Ondrej Miksik, Vibhav Vineet, Morten Lidegaard, Ram Prasaath, Matthias Nießner, Stuart Golodetz, Stephen L. Hicks, Patrick Perez, Shahram Izadi, and Philip H. S. Torr. The semantic paintbrush: Interactive 3d mapping and recognition in large outdoor spaces. In *ACM CHI*, 2015.

[151] Mohammadreza Mostajabi, Payman Yadollahpour, and Gregory Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *IEEE CVPR*, 2015.

[152] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE CVPR*, 2014.

[153] Michael C. Mozer. Backpropagation. *Complex systems*, 3(4):349–381, 1995.

[154] Feng Ning, Damien Delhomme, Yann LeCun, Fabio Piano, Léon Bottou, and Paolo Emilio Barbano. Toward automatic phenotyping of developing embryos from videos. *IEEE TIP*, pages 1360–1371, 2005.

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

[156] NVIDIA Corporation. CUDA Samples :: CUDA Toolkit Documentation, 2014.

[157] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L. Yuille. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. In *arXiv:1502.02734*, 2015.

[158] Devi Parikh and Kristen Grauman. Relative attributes. In *IEEE ICCV*, pages 503–510, 2011.

[159] Sylvain Paris and Fredo Durand. A fast approximation of the bilateral filter using a signal processing approach. *IJCV*, 81(1):24–52, 2013.

[160] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *ICML*, 2013.

[161] Deepak Pathak, Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional multi-class multiple instance learning. In *ICLR*, 2015.

[162] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *IEEE CVPR*, pages 2751–2758, 2012.

[163] Greg Shakhnarovich Payman Yadollahpour, Dhruv Batra. Discriminative re-ranking of diverse segmentations. In *IEEE CVPR*, 2013.

[164] Jian Peng, Liefeng Bo, and Jinbo Xu. Conditional neural fields. In *NIPS*, 2009.

[165] Tomas Pfister, James Charles, and Andrew Zisserman. Flowing convnets for human pose estimation in videos. In *IEEE ICCV*, 2015.

[166] Pedro H. O. Pinheiro and Ronan Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, 2014.

[167] Thomas K. Porter and Tom Duff. Compositing digital images. In *ACM SIGGRAPH*, pages 253–259, 1984.

[168] Renfrey Burnard Potts. Some generalized order-disorder transformations. In *Proceedings of the Cambridge Philosophical Society*, volume 48, pages 106–109, 1952.

[169] Brian L Price, Bryan Morse, and Scott Cohen. Geodesic graph cut for interactive image segmentation. In *IEEE CVPR*, pages 3161–3168, 2010.

[170] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *IEEE ICCV*, 2007.

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

[172] Jérôme Revaud, Philippe Weinzaepfel, Zaïd Harchaoui, and Cordelia Schmid. Deepmatching: Hierarchical deformable dense matching. *IJCV*, 2015.

[173] Christoph Rhemann, Carsten Rother, Jue Wang, Margrit Gelautz, Pushmeet Kohli, and Pamela Rott. A perceptually motivated online benchmark for image matting. In *IEEE CVPR*, 2009.

[174] Bernardino Romera-Paredes and Philip H. S. Torr. Recurrent instance segmentation. In *ECCV*, 2016.

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

[176] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio Lopez. SYNTHIA: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE CVPR*, 2016.

[177] Stephane Ross, Daniel Munoz, Martial Hebert, and J. Andrew Bagnell. Learning message-passing inference machines for structured prediction. In *IEEE CVPR*, 2011.

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

[179] Carsten Rother, Vladimir Kolmogorov, Yuri Boykov, and Andrew Blake. Interactive foreground extraction using graph cut. *Advances in MRF for Vision and Image Processing*, 2011.

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

[181] Jose C. Rubio, Joan Serrat, and Antonio López. Video co-segmentation. In *ACCV*, 2012.

[182] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Neurocomputing: Foundations of Research*, chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, 1986.

[183] Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 1999.

[184] Alexander G. Schwing and Raquel Urtasun. Fully connected deep structured networks. In *arXiv:1503.02351*, 2015.

[185] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014.

[186] Viktoriia Sharmanska, Novi Quadrianto, and Christoph H. Lampert. Augmented attribute representations. In *ECCV*, 2012.

[187] Yang Yu Sheng-Jun Huang and Zhi-Hua Zhou. Multi-label hypothesis reuse. In *KDD*, 2012.

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

[189] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Contour-based learning for object detection. In *IEEE ICCV*, 2005.

[190] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *IEEE CVPR*, 2011.

[191] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. Semantic texton forests for image categorization and segmentation. In *IEEE CVPR*, 2008.

[192] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost for image understanding: Multiclass object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81:2–23, 2009.

[193] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[194] Nathan Silberman, David Sontag, and Rob Fergus. Instance segmentation of indoor scenes using a coverage loss. In *ECCV*, 2014.

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

[196] Veselin Stoyanov, Alexander Ropson, and Jason Eisner. Empirical risk minimization of graphical model parameters given approximate inference, decoding, and model structure. In *AISTATS*, 2011.

[197] Paul Sturgess, Lubor Ladicky, Nigel Crook, and Philip H. S. Torr. Scalable cascade inference for semantic image segmentation. In *BMVC*, 2012.

[198] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. Image completion with structure propagation. *ACM TOG*, 24(3):861–868, 2005.

[199] Ilya Sutskever. *Training recurrent neural networks*. PhD thesis, University of Toronto, 2012.

[200] Charles A. Sutton, Andrew McCallum, and Khashayar Rohanimanesh. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *ICML*, 2004.

[201] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE CVPR*, 2015.

[202] Justin F Talbot and Xiaoqian Xu. Implementing grabcut. *Brigham Young University*, 2006.

[203] Meng Tang, Lena Gorelick, Olga Veksler, and Yuri Boykov. Grabcut in one cut. In *IEEE ICCV*, 2013.

[204] Sekhar C. Tatikonda and Michael I. Jordan. Loopy belief propagation and gibbs measures. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, 2002.

[205] James Thewlis, Shuai Zheng, Philip H. S. Torr, and Andrea Vedaldi. Fully trainable deep matching. In *BMVC*, 2016.

[206] Joseph Tighe and Svetlana Lazebnik. Understanding scenes on many levels. In *IEEE ICCV*, 2011.

[207] Joseph Tighe and Svetlana Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *IEEE CVPR*, 2013.

[208] Joseph Tighe and Svetlana Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. *IJCV*, pages 329–349, 2013.

[209] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *IEEE CVPR*, 1998.

[210] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014.

[211] Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Sharing features: Efficient boosting procedures for multiclass object detection. In *IEEE CVPR*, 2004.

[212] G. Tsoumakas, A. Dimou, E. Spyromitros-Xioufis, V. Mezaris, I. Kompatsiaris, and I. Vlahavas. Correlation-based pruning of stacked binary relevance models for multi-label learning. In *MLD 2009*, 2009.

[213] Zhuowen Tu. Auto-context and its application to high-level vision tasks. In *IEEE CVPR*, 2008.

[214] Zhuowen Tu, Xiangrong Chen, Alan L. Yuille, and Song Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. *IJCV*, pages 113–140, 2005.

[215] Jonas Uhrig, Marius Cordts, Uwe Franke, and Thomas Brox. Pixel-level encoding and depth layering for instance-level semantic labeling. In *arXiv preprint arXiv:1604.05096*, 2016.

[216] Jasper R. R. Uijlings, Koen E. A. van de Sande, Theo Gevers, and Arnold W. M. Smeulders. Selective search for object recognition. *IJCV*, pages 154–171, 2013.

[217] Koen van de Sande, Theo Gevers, and Cees G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9), 2010.

[218] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *ACM Multimedia*, 2015.

[219] Jakob Verbeek and William Triggs. Scene segmentation with CRFs learned from partially labeled images. In *NIPS*, 2007.

[220] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Joint optimization of segmentation and appearance models. In *IEEE ICCV*, pages 755–762, 2009.

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

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

[223] 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. *IJCV*, 2014.

[224] Vibhav Vineet, Jonathan Warrell, and Philip HS Torr. Filter-based mean-field inference for random fields with higher-order terms and product label-spaces. In *ECCV*, pages 31–44, 2012.

[225] Catherine Wah, Steve Branson, Pietro Perona, and Serge Belongie. Multiclass recognition and part localization with humans in the loop. In *IEEE ICCV*, 2011.

[226] Yang Wang and Greg Mori. A discriminative latent model of object classes and attributes. In *ECCV*, pages 155–168, 2010.

[227] Ronald J. Williams and David Zipser. Gradient-based learning algorithms for recurrent networks and their computational complexity, 1995.

[228] John Winn, Antonio Criminisi, and Thomas Minka. Object categorization by learned universal visual dictionary. In *IEEE ICCV*, 2005.

[229] Ralph Wolf and John C. Platt. Postal address block location using a convolutional locator network. In *NIPS*, 1994.

[230] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE CVPR*, pages 3485–3492, 2010.

[231] Jianxiong Xiao, James Hays, Bryan C Russell, Genevieve Patterson, Krista A Ehinger, Antonio Torralba, and Aude Oliva. Basic level scene understanding: categories, attributes and structures. *Frontiers in psychology*, 2013.

[232] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *IEEE ICCV*, 2015.

[233] Kaisheng Yao, Baolin Peng, Geoffrey Zweig, Dong Yu, Xiaolong Li, and Feng Gao. Recurrent conditional random field for language understanding. In *ICASSP*, 2014.

[234] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2015.

[235] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[236] Yimeng Zhang and Tsuhan Chen. Efficient inference for fully-connected crfs with stationarity. In *IEEE CVPR*, 2012.

[237] Ziyu Zhang, Sanja Fidler, and Raquel Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *IEEE CVPR*, 2016.

[238] Ziyu Zhang, Alexander G Schwing, Sanja Fidler, and Raquel Urtasun. Monocular object instance segmentation and depth ordering with cnns. In *IEEE CVPR*, 2016.

[239] Shuai Zheng, Ming-Ming Cheng, Wen-Yan Lin, Jonathan Warrell, Vibhav Vineet, Paul Sturgess, Nigel Crook, Nioly Mitra, and Philip H. S. Torr. ImageSpirit: Verbal Guided Image Parsing. *ACM TOG*, 2014.

[240] Shuai Zheng, Ming-Ming Cheng, Jonathan Warrell, Paul Sturgess, Vibhav Vineet, Carsten Rother, and Philip .H. S. Torr. Dense semantic image segmentation with objects and attributes. In *IEEE CVPR*, 2014.

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

[242] Shuai Zheng, Victor Adrian Prisacariu, Melinos Averkiou, Ming-Ming Cheng, Niloy Mitra, Jamie Shotton, Philip H. S. Torr, and Carsten Rother. Object proposal estimation in depth images using compact 3D shape manifolds. In *German Conference on Pattern Recognition*, 2015.

[243] Shuai Zheng, Paul Sturgess, and Philip H. S. Torr. Approximate structured output learning for constrained local models with application to real-time facial feature detection and tracking on low-power devices. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2013.

[244] Youyi Zheng, Xiang Chen, Ming-Ming Cheng, Kun Zhou, Shi-Min Hu, and Niloy J. Mitra. Interactive images: Cuboid proxies for smart image manipulation. *ACM Trans. Graph.*, 31(4):99:1–11, 2012.

[245] Shizhe Zhou, Hongbo Fu, Ligang Liu, Daniel Cohen-Or, and Xiaoguang Han:. Parametric reshaping of human bodies in images. *ACM Trans. Graph.*, 29(4):126, 2010.

[246] Jun-Yan Zhu, Jiajun Wu, Yichen Wei, Eric Chang, and Zhuowen Tu. Unsupervised object class discovery via saliency-guided multiple class learning. In *IEEE CVPR*, pages 3218–3225, 2012.