

UCLA

UCLA Electronic Theses and Dissertations

Title

Joint Multiple Visual Task Understanding from a Single Image via Deep Learning and Conditional Random Field

Permalink

<https://escholarship.org/uc/item/2s11c5fx>

Author

Wang, Peng

Publication Date

2017

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Joint Multiple Visual Task Understanding from a Single Image via Deep Learning and Conditional
Random Field

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Statistics

by

Peng Wang

2017

© Copyright by

Peng Wang

2017

ABSTRACT OF THE DISSERTATION

Joint Multiple Visual Task Understanding from a Single Image via Deep Learning and Conditional
Random Field

by

Peng Wang

Doctor of Philosophy in Statistics

University of California, Los Angeles, 2017

Professor Alan Loddon Yuille, Chair

Human are interpolating the visual world with very rich understanding. For example, when observing the world through eyes, we not only understand the high level semantic meaning of each region/pixel, more importantly, we also understand the 3D properties like how far away each object is and how the 3D shape of each object is in order to do interaction with the world. In the field of computer vision, however, visual understanding are separated into multiple tasks, e.g. segmentation, 3D reconstruction or object detection etc., due to its high complexity. However, this induces the problem that the results from different strategies are lack of compatibility among different tasks. For example, semantic object detection can not take care of the 3D occlusion regions, while 3D reconstruction does not consider overall semantic context. Thus, in order to have good visual understanding, it is critical to joint understand different tasks while maintaining their compatibility. Luckily, thanks to the raising technique of deep learning, (a.k.a. convolutional neural network (CNN)), which dramatically beats the other traditional strategies in many visual tasks based on hierarchical learned features with a nearly single framework, we are able to unify different understandings in a more compact and efficient way by designing reasonable output and interaction terms. However, CNN is not a magic key of solving all problems, and one obvious limitation of CNN is that it contains arbitrarily selected convolutional kernel size and layers, yielding non-adaptive receptive fields to match the variance of object scales. In addition, it is not strait-forward to add

arbitrary connections inside each layer based on intuition. Thus, we further embed the conditional random field (CRF) into the system in order to compensate the deficiency in order to unify different cues and perform multiple tasks simultaneously.

In this thesis, we prove the concept through estimating multiple tasks jointly including joint part and object segmentation, joint segmentation and geometry estimation etc. We first show that we can fit deep convolutional network into many different tasks to acquire superior performance compare to traditional shallow features. Secondly, by unifying different tasks with our designed compatibility constrains, we make different tasks mutually regularized and beneficial. Finally, to evaluate the results, we perform our experiments over the standard evaluating benchmarks like PASCAL [EVWa] for segmentation and the NYU v2 dataset [SHK12] for depth estimation. Last but not the least, we not only apply the existing metrics to show the performance gain from our design, but also introduce reasonable new metrics in order to better show the aspect that improved.

The dissertation of Peng Wang is approved.

Demetri Terzopoulos

Hongjing Lu

Yingnian Wu

Alan Loddon Yuille, Committee Chair

University of California, Los Angeles

2017

TABLE OF CONTENTS

1	Introduction	1
1.1	Motivation & Problem Statement	1
1.2	Contributions	2
1.3	Thesis outline	3
2	Preliminaries	4
2.1	Visual tasks in this thesis	4
2.2	Fully Convolutional Network (FCN)	5
2.3	Conditional Random Field (CRF)	6
3	Deep occlusion estimation from a single image	8
3.1	Related work	9
3.2	The DOC deep network	11
3.2.1	Occlusion relations: Representation and Loss functions	11
3.2.2	The network architecture	14
3.3	Pascal instance occlusion dataset (PIOD)	16
3.4	Experiments	19
3.4.1	Evaluation criteria	20
3.4.2	Performance comparisons.	21
3.5	Conclusion and future work	26
3.6	Appendix	27
4	Joint segmentation and 3D geometry estimation from a single image	30
4.1	Related work	31

4.2	Overview	32
4.3	DCRF for Surface Regularized Geometry Estimation	33
4.3.1	Unary terms	33
4.3.2	Pairwise term for regularization.	34
4.4	Optimization	36
4.4.1	Inference	36
4.4.2	Joint training of CNN and DCRF	37
4.5	Implementation details for DCRF	38
4.6	Experiments	39
4.7	Conclusion	44
4.8	Appendix for SURGE	45
4.8.1	Visualization of network prediction confidence	45
4.8.2	Details of extracting ground-truth planar surfaces	46
4.8.3	Network structures for plane and edge prediction.	46
4.8.4	Learning, inference time	46
4.8.5	Additional qualitative comparison from NYU v2 data.	47
5	Joint part and object segmentation with deep learned potentials	51
5.1	Related work	53
5.2	Joint part and object segmentation	55
5.2.1	Semantic compositional parts	55
5.2.2	Deep part and object potentials	56
5.2.3	SCP segments proposal	57
5.2.4	Joint FCRF	58
5.2.5	Relation to the DPM structure	60

5.3	Experiments	61
5.3.1	Model training.	61
5.3.2	Parameters and details	63
5.3.3	Performance comparisons	64
5.4	Conclusion and discussion	70
6	Discussion, conclusion and future directions	72
	References	74

LIST OF FIGURES

2.1	The fully convolutional network (FCN) proposed by Long et.al [LSD15b].	5
2.2	Comparison results between the commonly used grid CRF (up row) and the fully connected CRF proposed by Philipp et.al [KK12a] (bottom row).	6
3.1	Left: Occlusion boundaries represented by orientation θ (the red arrows) which indicates occlusion relationship using the “left” rule where the left side of the arrows is foreground. Right: More examples from our Pascal instance occlusion dataset (PIOD).	10
3.2	Four ways to represent occlusion relations, see Sec. 3.2.1. Consider the boundary contour joining triple points 1 and 3 where border ownership changes at the midpoint 2. The background is to the left of the border between points 1 and 2, but it is on the right of the border between 2 and 3. The “left” rule uses the occlusion orientation on the contour, see arrow, to indicate border ownership (i.e. the left side of the arrow specifies the side of foreground). In panels (a) and (b), the triples points 1, 3 and the junction 2 are explicitly represented. Panel (a), see [RFM06], uses binary variables e_{12}, e_{23} to specify border ownership, while panel (b) includes explicit semantic knowledge [HEH11] where regions are represented by their semantic types, e.g., t_1, t_2 . In panels (c) and (d), the representation is pixel-based. $\mathbf{L}_{o,q}, \mathbf{L}_o$ represent occlusions in terms of a boundary indicator and an occlusion orientation variable (the dense arrows) using the “left” rule to indicate border ownership (i.e. which side is foreground). The difference is that in panel (c), see [TFA15], $\mathbf{L}_{o,q}$ quantizes the occlusion orientation to take 8 values. In contrast, in panel (d), $\mathbf{L}_o$ allows it to take continuous values.	12
3.3	The orientation probability $P(\theta_j^* \theta_j, \mathbf{W})$ as a function of the difference between the predicted and ground truth orientation, i.e. θ^* and θ in the figure.	13

3.4	Top: We explored two alternative DOC architectures for occlusion recovery: (i) HED, and (ii) DMLFOV. Bottom: for inference, we first apply a two stream network (shown for HED) to predict pixel-wise boundaries and the occlusion orientations respectively. Then, we apply non-maximum suppression (NMS) to the boundaries, merge the two predictions, and recover the occlusion boundaries.	15
3.5	The annotation process of our PIOD. Given an image, we provide two annotated maps, i.e. (b) the semantic instance map and (c) the generated boundary map. An annotator needs to supplement the boundary map with directed line segments following the “left” rule. We assume the objects occlude background by default, so the annotator only needs to label the boundaries violating this rule or between adjacent instances. Finally, we match the labelled line segments to all the boundaries as shown in (d).	17
3.6	The occlusion relationship matrix shows the frequency of occlusion between different classes (red means high). The horizontal axis denotes the occluded objects.	18
3.7	Center: the two ground truth maps for each image in the BSDS ownership data. Left: limitation of the orientation map generated by SRF-OCC [TFA15] for occlusion evaluation. In the white rectangle, the white arrows show the quantized ground truth orientation at corresponding pixels, which is not smooth or intuitively correct. Also, at bottom right, label inconsistent ground truth edges are discarded when fusing multiple maps. Right: our proposed multiple ground truth occlusion orientation maps for learning and evaluation.	20
3.8	Quantitative comparison on BSDS ownership data. SRF-OCC [TFA15] is the baseline model. In (b), we show the edge detection performance on BSDS 500 testing data with models trained from the BSDS ownership data (with only 100 images). This shows the DOC-HED model we trained are comparable to those in the HED paper (best viewed in color). Details are in Sec. 3.4.2.	22
3.9	Quantitative comparison over our PIOD data.	23

3.10	Qualitative comparisons (best viewed in color). At the left side of each column, we show algorithm results compared with ground truth. The “red” pixels with arrows are correctly labelled occlusion boundaries, , the “green” pixels are correctly labeled boundaries but incorrect occlusion, and the “blue” pixels are false positive boundaries. At the right of each column, we show the occlusion boundaries by a 2.1D relief sculpture. In the figure, the foreground regions are raised (embossed). (a) Comparisons on the BSDS ownership data between SRF-OCC [TFA15] and DOC-HED. (b) Comparisons on PIOD between DOC-HED and DOC-DMLFOV. Note for some images, some internal occlusion boundaries are recovered (although they are not labelled correct), e.g., the tire on the bike and the woman’s right arm. This that DOC can generalize from boundaries to some internal edges. We give more examples in Fig.1 of the supplementary material.	25
3.11	Qualitative comparison examples over the BSDS border ownership data (Best view in color).	28
3.12	Qualitative comparison examples over the PIOD (Best view in color).	29
4.1	Framework of SURGE system. (a) We induce surface regularization in geometry estimation though DCRF, and enable joint learning with CNN, which largely improves the visual quality (b).	31
4.2	The orthogonal compatibility constraint inside the DCRF. We recover 3d points from the depth map and require the difference vector to be perpendicular to the normal predictions.	34
4.3	Pairwise surface affinity from the plane and edge predictions with computed Ncut features. We highlight the computed affinity w.r.t. pixel i (red dot).	36
4.4	Four types of ground-truth from the NYU dataset that are used in our algorithm.	39
4.5	Visual comparison between network output from Eigen et.al [EF15] and our results in 3D view. We project the RGB and normal color to the 3D points (Best view in color).	44

4.6	Example to visualize of the prediction variation which can be used to represent prediction confidence by using the Monte Carlo dropout (last column). We highlighted place where confidence indicate errors for depth and normal.	45
4.7	Architecture of edge network with two input. At the corner. of each layer we show the number channel x the number of sub-layers.	47
4.8	Architecture of plane network that takes RGB image and output a binary plane map.	47
4.9	Additional visual comparison between network output from Eigen et.al [EF15] and our results (Best view in color) from NYU v2 dataset.	48
4.10	Additional visual comparison between network output from Eigen et.al [EF15] and our results (Best view in color) from NYU v2 dataset.	49
4.11	Additional visual comparison between network output from Eigen et.al [EF15] and our results (Best view in color) from NYU v2 dataset.	50
5.1	We handle the prediction of semantic object and part segmentation in a wild scene scenario. (a) Original image. (b)&(c) are the object and part segmentation respectively, generated from our algorithm.	52
5.2	Our framework for joint part and object segmentation. Given an image, a two-stream FCN is performed to predict both the semantic compositional parts (SCP) and object potentials with different input image scales. The two potentials are then concatenated and fed to a new convolutional layer to predict joint object potentials. Finally, from SCP potentials, SCP segments are proposed as nodes for the fully-connected CRF to jointly infer the part and object labels.	53
5.3	Illustration of our semantic compositional part (SCP) grammer in Sec. 5.2.1. Each SCP is associated with one semantic meaning, and all the objects are composed of several SCPs.	56

5.4	The SCP segments proposals handle various difficulties such as truncation, occlusion, deformation and view point changes. The colors maps shown at the bottom for each kind of SCP. We keep the color consistent in the results and will show more in our supplementary materials.	58
5.5	Investigation of image scales. Top: an example from our joint object FCN prediction, showing that larger scale leads to finer boundaries, but introduces local ambiguities. Bottom: validate the accuracy of scale combination of s_o and s_p	63
5.6	Comparison examples with SPS [WY15] and HC [HAG15] in the Horse-Cow dataset and the color map is in Fig. 5.4 (Best view in color).	66
5.7	Comparison examples with FCN [LSD15b] for object segmentation and HC* [HAG15] for part segmentation in the Quadrupeds dataset. The object label color map is shown at above and part label color map is shown in Fig. 5.4 (Best view in color). .	68

LIST OF TABLES

4.1	Normal accuracy comparison over the NYU v2 dataset. We compare our final results (DCRF-conf (JT)) against various baselines over ground truth planar regions at upper part, where JT means joint training CNN and DCRF as presented in Sec. 4.4.2. We list additional comparison over the edge and full image region at lower part. . .	42
4.2	Depth accuracy comparison over the NYU v2 dataset.	43
5.1	Average precision over the Horse-Cow dataset.	65
5.2	Average precision over the Quadrupeds data.	67
5.3	Average precision over the part segmentation benchmark.	70

ACKNOWLEDGMENT

I am sincerely thankful to many people for their help and support along the way till the Ph.D. Firstly, I appreciate Professor Alan Yuille, who adopt me into his group after I applied for US universities for a long time. He is a knowledgeable teacher, a great thinker and researcher in this field with his unique opinion about the development. Different from most young professors in this field, who require the publication of papers without thinking about the fundamentals of vision but busy in propose new tasks, he keens to understanding the basic properties of vision, such as hierarchical model, 3D and symmetry. He never forces me to only work in his way, but discusses with us a lot friendly and provides intuitions. After I study in this field 6 years, I just notice the important role in solving the problems. Though I worked on few of them such as 3D following the trend of deep learning, I still having the core idea of agreement with Alan about the fundamentals. In addition, to my thesis defense committee, Yingnian Wu, Hongjing Lu, and Demetri Terzopoulo, I appreciate for their feedback and advice of this work, and also for their valuable time taken to participate in my defense.

Here I would like to thank my collaborators, Xiaohui Shen, Zhe Lin, Scott Cohen, Brain Price and Bryan Russell from Adobe, who helped me during my internship, correct the directions from high-level and support the writing of papers at publications. We are a group of hard workers but enjoy the friend relationships. I appreciate the collaborate from my labmates, Fangting Xia who spend a lot time experimenting the idea we developed. Xianjie Chen we provide me support of models and datasets. Liang-Chieh Chen who support us with the results and code. I also thank Lingxi Xie, Zhou Ren for carefully revising my papers. Also other labmates, Xiaochen Lian, Junhua Mao, Jianyu Wang, Weichao Qiu and Chenxi Liu who provide me friendly environment to play games and do research. This will be a unforgettable group in my life.

Finally, I want to thank my family, my mom Xumei Meng give me warm support and carefulness along the way, and my dad Jinquan Wang guided me with his life wisdom. At last, I appreciate my wife, Si Cheng, who give me her love even though we are apart now with a distance of thousands of miles. She also give up her position of doctor at Beijing, and come to US achieving a future with me finally after four years. I thank with my heart to her for the braveness and accompany.

VITA

- 2006-2010 Bachelor Degree in the Department of Machine Intelligence, Peking University.
- 2010-2013 Master Degree in the Department of Machine Intelligence, Peking University.
- 2011-2016 Ph.D Candidate in the Department of Statistics, UCLA.
Research Assistant in Center For Cognition, Vision, and Learning, UCLA.

PUBLICATIONS

- Joint Multi-Person Pose Estimation and Semantic Part Segmentation in a Single Image.* Fangting Xia, **Peng Wang**, Alan Yuille, In CVPR (2017)
- SURGE: Surface Regularized Geometric Estimation from a Single Image.* **Peng Wang**, Xiaohui Shen, Bryan Russel, Scott Cohen, Brian Price, Alan Yuille, In NIPS (2016)
- Zoom Better to See Clearer: Human and Object Part Segmentation with Auto Zoom Net.* Fangting Xia, **Peng Wang**, Liang-Chieh Chen, Alan Yuille, ECCV (2016)
- DOC: Deep OCclusion Recovering From A Single Image.* **Peng Wang**, Alan Yuille, ECCV (2016)
- Joint Object and Part Segmentation using Deep Learned Potentials.* **Peng Wang**, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, Alan Yuille, In ICCV (2015)
- Pose-Guided Human Parsing with Deep Learned Features.* Fangting Xia, **Peng Wang***, Jun Zhu*, Alan Yuille, In AAAI (2016) (* indicates equal contribution)
- Towards Unified Depth and Semantic Prediction from a Single Image.* **Peng Wang**, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, Alan Yuille, In CVPR (2015)
- Error Factor Analysis for Wild-scene Image Labelling.* **Peng Wang**, Alan Yuille, In WACV, (2015)

CHAPTER 1

Introduction

Making the computer perceive the world through color camera with similar understanding as human beings is the ultimate goal of computer vision. While due to the high complexity of the content from images, researchers in computer vision factorize the visual understanding into multiple single visual tasks, from the low-level edge detection [AMF11, LSS12], texture patterns [MFM04], mid-level geometry understanding [SSN09], part understanding [BM09a], to the high-level semantic object and scene understanding [KSH12, GDD14, LSD15b], yielding different algorithm targeting at specific tasks without much considering the others. In addition, those visual tasks can also be applied to many practical image processing problems, e.g. super resolution [LTH16], image deblur [XRL14] and image in-painting [XXC12] etc.

However, human are perceiving the visual world with joint estimation of all of the tasks simultaneously, especially without conflicting about the understanding of multiple visual problems. For example, by looking at an indoor room, when parsing a region as floor to walk on, we expect the the region to be flat and facing up in 3D, which means that we are not allow some region that is vertical to be understood as floor region. The consistency through different visual tasks in either low-level, mid-level or high-level understanding leading us to non-confusing interaction with the world. Thus, in computer vision, algorithms should not cut the connection between different visual tasks and should reasonably have the right constrains.

1.1 Motivation & Problem Statement

With the raising of the application of machine learning into computer vision tasks, fusing different task together is not a new idea. Back the the date of combining SIFT and HOG features [Low04,

DT05] with the support vector machine and conditional random fields [LMP01], researchers have already trying to combining visual task together to improve the robustness of each individual tasks, such as joint semantic parsing and 3D reconstruction based on multiple view images [SFP13, KLD14], object detection and parsing [TCY05, YFU12, DCY14], and joint pose estimation and human parsing [DCS14] etc. However, due to high dimension of the output space in the graphical model constructed, the inference and learning speed are slow with those old features, especially when the inference is over pixel accuracy. In addition, the tasks have already tried in the previous works are limited in terms of the number of tasks.

In this thesis, firstly, we bring in the convolutional neural network (CNN) which could vastly use the parallel computational ability of graphic processing unit(GPU) to speed up the processing, in addition, due to the deep hierarchy of the features, the model has extremely high capacity to learn the pattern embedded in large amount the data, yielding superior performance comparing to the traditional methods. The deep network has been spread over and dominant the performance of most tasks as we mentioned before. In addition, there even appear a super network which have multiple branched outputs to handle most of existing tasks [Kok16b]. Secondly, depend on CNN, recently there appears a few works try to unified different information effectively, e.g. joint edge and segmentation [BST16], joint optical flow and segmentation [SSJ16], detection and segmentation [HAG15]. In our case, we try to extend the joint inference additional to the ones already tried before, including occlusion edge detection [WY16], semantic part and object segmentation [WSL15a], geometric region segmentation and reconstruction from a single image [WSL15a, WSR16]. From those works, we find the pixel-wise inferred results from CNN is dramatically better than traditional methods and by additionally adding joint constrains between different visual tasks, the final system always gives performance improvements.

1.2 Contributions

In summary, the contribution of this thesis includes three aspects:

1. We encouraged the field to not only focus on a single visual task and over-fit each individual dataset, but also trying to design a "full stack" vision algorithm with powerful understanding

of multiple visual tasks jointly. We claim that only by doing so, the vision system could be complete and robust, and we prove such concept with several practical cases.

2. In order to efficiently and effectively combine different tasks, we induce CNN and CRF in our system by designing our own structure of the networks and terms of the graphical model, which proved to be effective in our experiments and out-perform the previous state-of-the-art works.
3. Finally, for better evaluation of the system, we construct new dataset and propose additional new evaluation metrics in most of our works. This is especially important because firstly, it is necessary to have a benchmark to quantitatively understand the overall performance of different systems. In addition, an overall single metric can hardly reveal the pros and cons of an algorithm, thus metrics focusing on different aspects are important for us to interpolate and improve the algorithms.

1.3 Thesis outline

The thesis is organized as following. In Chapter 2, we will introduce the key technique used in our works including fully convolutional networks (FCN) and conditional random field (CRF) with unary terms and pair-wise terms for inference. In Chapter 3, we show how the FCN model is applied to a classical occlusion boundary inference by designing an orientation representation and proper loss function. In addition, we constructed a new occlusion dataset based on the popular PASCAL images [EVWa] by providing additional labels to the pixels over instance edges. In Chapter 4, we show how to combine FCN model and CRF to do joint segmentation and 3D geometry estimation by introducing a network with multiple branches or streams of output and multiple losses, given the multiple output we design pairwise terms between geometry and segments to enforce the consistency between different information. Chapter 5 shows another case by combining multiple level of segmentation, i.e. part level segmentation and object segmentation, to achieve a hierarchical understanding of object regions in an image. Finally, in Chapter 6, we concludes this thesis and show promising direction of multiple task learning and inference.

CHAPTER 2

Preliminaries

In this chapter, we provide some preliminaries for the thesis. Firstly, we will introduce and formulate the input and output of visual tasks included in our model. Then, we will briefly explain the FCN and CRF which serve as our main techniques.

2.1 Visual tasks in this thesis

All of our visual tasks are considering the input to be an 2d image, i.e. $\mathbf{I}$ which is a 3D matrix $h \times w \times 3$, 3 is the channel of RGB. For different tasks we require different kinds of output.

Occlusion Boundary Detection. The output contains two label maps, and the first one is a 2d $h \times w$ matrix $\mathbf{E}$, where the value at each pixel is a binary number with 0 representing a non-edge pixel and 1 representing an edge pixel. The second label map is an orientation map, which is a 2d $h \times w$ matrix $\mathbf{O}$ where the value at each pixel is a continuous number from 0 to 2π , representing an angle direction, and the left side of the direction indicates foreground as show in Figure 3.1.

Geometry Estimation. The output is generally a depth map represented as a 2d $h \times w$ matrix $\mathbf{D}$ with continuous value, where the value at each pixel is the 3d distance to the camera. Additionally, the geometry may also include a normal map represented as a 3d $h \times w \times 3$ matrix $\mathbf{N}$ with continuous value, where the 3d vector at each pixel is a unified vector representing the 3d normal direction.

Semantic segmentation. The output is a 2d $h \times w$ label map $\mathbf{L}$ with discrete value at each corresponding pixel, representing the semantic label id of each pixel. Notice that we can ask for multiple outputs when there are multiple label set. For example, when we do hierarchical

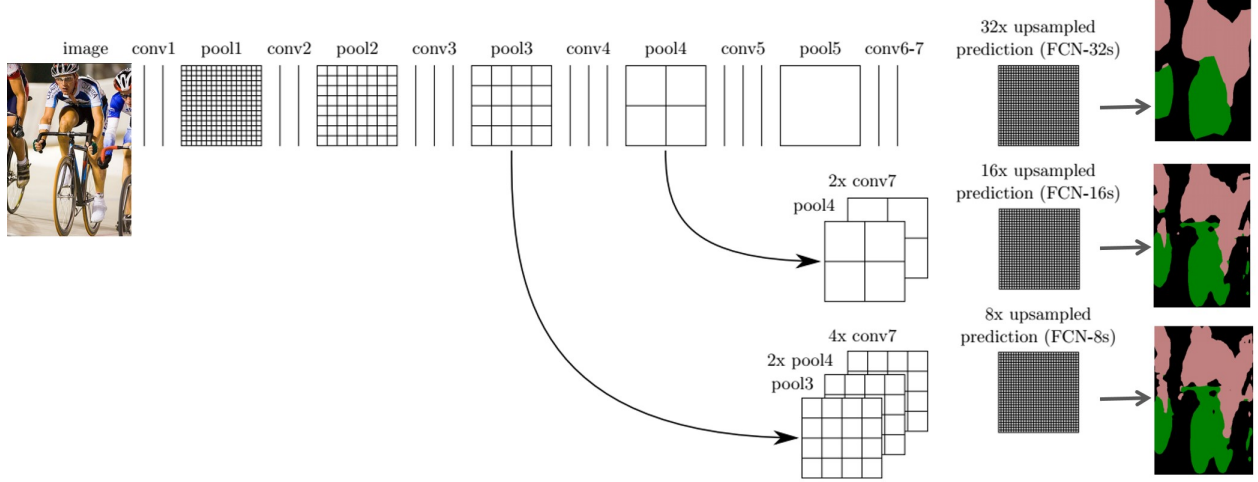


Figure 2.1: The fully convolutional network (FCN) proposed by Long et.al [LSD15b].

segmentation, i.e. segment an image into multiple level of semantics, such as object and object part segmentation, the output should include multiple label maps.

2.2 Fully Convolutional Network (FCN)

From the first time of deep convolutional network, known as AlexNet [KSH12], won the ImageNet competition [DDS09], the network structure are progressed vastly during the last four years including VGG network [SZ14], Residual network [HZR16], etc. The FCN is a special case of deep network, where the fully connected layer are removed and transformed to convolutional operation. It relaxes the input image size for the network to be not limited as a fixed size, e.g. 224×224 for Alex net. Thus, rather than just outputting a single label from each forward, the network can yield the dense prediction at each pixel with a fixed context. This type of network is firstly applied in semantic segmentation on PASCAL dataset [EVWb] by Long et.al. In Figure 2.1, we show a basic type of the FCN network used in [LSD15b]. We notice that the output from the last layer is coarse and blurry at boundary regions, which is majorly due to the pooling operation in the network drops detailed edge information. Long et.al try to solve the problem by induce skip connection from previous layers and average all the outputs together, yielding result with finer details.

Later, in order to include larger context, additional networks are further developed, such the

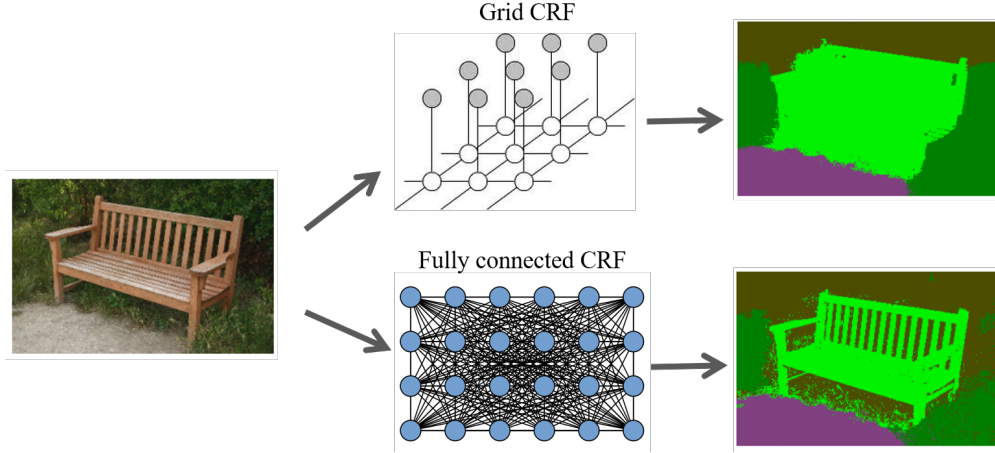


Figure 2.2: Comparison results between the commonly used grid CRF (up row) and the fully connected CRF proposed by Philipp et.al [KK12a] (bottom row).

encoder-decoder network [NHH15], multi-scale network [EF15], the hourglass network [NYD16] etc. In term of tasks, it is also further developed to deal with edge detection [XT15], depth and normal prediction [EF15], human pose estimation [NYD16] etc, which yields superior performance than traditional methods.

2.3 Conditional Random Field (CRF)

CRF [LMP01] is a classical graphical model, which has been used in the field of computer vision for about 20 years, and known for its flexibility of designing the relationship between different information and the connection between neighboring context. In this thesis, we majorly care about doing inference through learned potentials, while not pay attention to a joint learning with the normalize factor. Thus, by taking log likelihood, the formulation of CRF with only pairwise connection is generally represented as,

$$\min_{\mathbf{y}} \sum_i \phi(y_i | \mathbf{x}_i) + \lambda \sum_{i,j} \psi(y_i, y_j | \mathbf{x}_i, \mathbf{x}_j),$$

where $\mathbf{y}$ is the label space of the graph, and $\mathbf{x}_i$ is the feature for each node i . For each node, it contains a unary term and a pairwise term, where the unary term measures the energy of the

optimized label y_i based on its own feature $\mathbf{x}_i$, while the pairwise term measures the energy from its connected neighbor nodes $\{j | (j, i) \in \mathcal{E}\}$. Most commonly, the pairwise term for each node only contains connections to the spacial adjacent nodes for smoothness and regularization. This is due to large amount of connection would result in high complexity in inference. While we are more interested in fully connected CRF [KK12a], as illustrated in Figure 2.2, where every pair of node is connected. In the work, they propose to use Gaussian kernels and permutohedral lattice to reduce the computational complexity. This makes the graph possible to acquire long range context for decision, yielding more robust results than the adjacent cases.

Given the potential, for inference, we may adopt many inference algorithms such as loop belief propagation, mean field approximation etc. based on the type of graph we constructed. In this framework, the key to successfully apply CRF is properly design the pairwise potentials, through firstly well understand the task given and relationship we can intuitive extract from different cues, and then formulate the intuition as simple as possible for fast learning and inference. We will illustrate this through the cases later.

CHAPTER 3

Deep occlusion estimation from a single image

Humans are able to recover the occlusion relationships of objects from single images. This has long been recognized as an important ability for scene understanding and perception [Gib68, Bie81]. As shown on the left of Fig. 3.1, we can use occlusion relationships to deduce that the person is holding a dog, because the person’s hand occludes the dog and the dog occludes the person’s body. Electrophysiological [HMQ05] and fMRI [FBK09] studies suggest that occlusion relationships are detected as early as visual area V2. Biological studies [CSN07] also suggest that occlusion detection can require feedback from higher level cortical regions, indicating that long-range context and semantic-level knowledge may be needed. Psychophysical studies show that there are many cues for occlusion including edge convexity [KG76], edge-junctions, image contrast, intensity gradients, and texture [PG08].

Computer vision researchers have also used similar cues for estimating occlusion relations. A standard strategy is to apply machine learning techniques to combine cues like convexity, triple-points, geometric context, image features like HOG, and spectral features, e.g. [RFM06, HEH11, TFA15]. These methods, however, mostly rely on hand-crafted features and have only been trained on the small occlusion datasets currently available. But in recent years, fully convolutional deep convolutional neural networks [LSD15b] which exploit local and non-local cues, and which are trained on large datasets, have been very successful for related visual tasks such as edge detection [XT15] and semantic segmentation [CPK15]. In addition, visualization of deep networks [ZF14, MV15] show that they can also capture and exploit the types of visual cues needed to estimate occlusion relations.

This motivates us to apply deep networks to estimate occlusion relationships which requires constructing a large annotated occlusion dataset. This also requires making design choices such

as how to represent occlusion relations and what type of deep network architecture is best able to capture the local and non-local cues required. We represent occlusion relations by a per-pixel representation with two variables: (i) a binary edge variable to indicate if a pixel is on a boundary, and (ii) a continuous-valued occlusion orientation variable (at each edge pixel) in the tangent direction of the edge whose direction indicates the occlusion relationship using the left rule (i.e. the region to the left of the edge is in front of the region to the right). Our DOC network architecture is based on recent fully convolutional networks [LSD15b] and is multi-scale so that it can take into account local and non-local image cues. More specifically, we design two versions of DOC based on [XT15] and [CPK15] respectively.

To construct our dataset, we select PASCAL VOC images [EVW10] where many of the object boundaries have already been annotated [HAB11, CML14]. This simplifies our annotation task since we only have to label the occlusion orientation variable specifying border ownership. Our Pascal Instance Occlusion Dataset (PIOD) consists of 10,000 images and is two orders of magnitude larger than existing ones such as the BSDS border ownership [RFM06] (200 images) and GeoContext [HEH11] (100 images). We note that the NYU depth dataset [SHK12] (1449 indoor images) can also be used to test occlusion relations, but it is restricted to indoor images.

This paper makes two main contributions: (1) We design a deep network architecture to estimate occlusion relations and show that it outperforms state-of-the-art methods (by at least 5AP) and is computationally efficient (0.6s/image). (2) We create a large occlusion boundary dataset over the PASCAL VOC images, which is a resource for studying occlusion. We will release all our models, codes and dataset upon the publication of this paper.

3.1 Related work

In computer vision, studying occlusion relations has often been confined to multiview problems such as stereo and motion [SH09, HY10, SBM11, ARS12, WRH15]. In these situations multiple images are available and so occlusion can be detected by finding pixels which have no correspondence between images [GLY92, BM92].

Inferring occlusion relations from a single image is harder. Early work restricted to simple



Figure 3.1: Left: Occlusion boundaries represented by orientation θ (the red arrows) which indicates occlusion relationship using the “left” rule where the left side of the arrows is foreground. Right: More examples from our Pascal instance occlusion dataset (PIOD).

domains, e.g. blocks world [Rob63] and line drawings [Coo97] using a variety of techniques ranging from algebraic [Sug84] to the use of markov random fields (MRF) for capturing non-local context [Sau05, YLK01]. The 2.1D sketch [NM90] is a mid-level representation of images involving occlusion relations, but it was conceptual and served to draw attention to the importance of this task.

Research on detecting occlusion relations in natural images was stimulated by the construction of the BSDS border ownership dataset [RFM06]. Computer vision methods typically addressed this problem using a two stage approach. For example, [RFM06] used the Pb edge detector [MFM04] to extract edge features and then used a markov random field to determine foreground and background. This was followed up [LL09] who used a richer set of occlusion cues. Other work by [HEH11] introduced the use of explicit high-level cues including semantic knowledge (e.g., sky and ground) and introduced a new dataset GeoContext for this purpose. Note that in this paper we do not use explicit high-level cues although these might be implicitly captured by the deep network. Most recently, [TFA15] used multiple features (e.g., HOG) combined with structure random forest (SRF) [DZ15] and geometric grouping cues (for non-local context) to recover the boundaries and foreground background simultaneously.

Occlusion relations can also be addressed using techniques which estimate 3D depth from single images. These methods typically use either MRF (to capture non-local structure) [HEH07, SSN09, LGK10], deep learning [EF15], or combinations of both [WSL15a, LSL15b, LSD15a]. These studies do not explicitly attempt to estimate occlusion, but it can be deduced by detecting the depth discontinuities in the estimated depth map. To train these methods, however, requires annotated

3D data which is hard to obtain for outdoor images, such as those in PASCAL VOC. Hence these methods are most suitable for indoor studies, e.g., on the NYU depth dataset [SHK12].

Our method builds on the fully convolutional network literature and, in particular, recent work on edge detection [XT15] and semantic segmentation [CPK15] which exploit multi-scale and capture local and non-local cues. We also handle network downsampling by combining the "hole" algorithm [CPK15] and deconvolution [LSD15b].

3.2 The DOC deep network

This section describes our DOC deep network. Designing this network requires addressing two main issues: (1) specifying a representation for occlusion relations and a loss function, (2) a deep network architecture that captures the local and non-local cues for detecting occlusion. We now address these issues in turn.

3.2.1 Occlusion relations: Representation and Loss functions

Representing occlusion relations. We represent occlusion relations using an edge map to represent the boundaries between objects (and background) and an orientation variable to indicate the depth ordering across the boundary. We first review existing methods for representing occlusion to motivate our choice and clarify our contribution.

Methods for representing occlusion relations can be roughly classified into four types as shown in Fig. 3.2. The first two types, panels (a) and (b), represent triple points and junctions explicitly (we defined junctions to be places where border ownership changes). The third and fourth types, panels (c) and (d) use a pixel-based representation with a pair of label indicating boundary and occlusion orientation. The representations in panels (a) and (b) were used in [RFM06] and [HEH11] respectively. A limitation of computer vision models which uses these types of representations is that performance is sensitive to errors in detecting triple points and junctions. The representation in panel (c) enables the use of pixel-based methods which are more robust to failures to detect triple points and junctions [TFA15]. But it quantizes the occlusion orientation variable into 8 bins, which

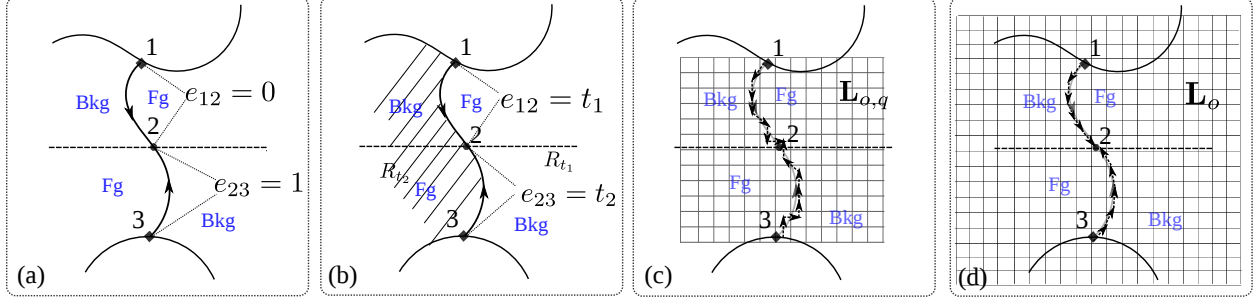


Figure 3.2: Four ways to represent occlusion relations, see Sec. 3.2.1. Consider the boundary contour joining triple points 1 and 3 where border ownership changes at the midpoint 2. The background is to the left of the border between points 1 and 2, but it is on the right of the border between 2 and 3. The “left” rule uses the occlusion orientation on the contour, see arrow, to indicate border ownership (i.e. the left side of the arrow specifies the side of foreground). In panels (a) and (b), the triple points 1, 3 and the junction 2 are explicitly represented. Panel (a), see [RFM06], uses binary variables e_{12}, e_{23} to specify border ownership, while panel (b) includes explicit semantic knowledge [HEH11] where regions are represented by their semantic types, e.g., t_1, t_2 . In panels (c) and (d), the representation is pixel-based. $L_{o,q}, L_o$ represent occlusions in terms of a boundary indicator and an occlusion orientation variable (the dense arrows) using the “left” rule to indicate border ownership (i.e. which side is foreground). The difference is that in panel (c), see [TFA15], $L_{o,q}$ quantizes the occlusion orientation to take 8 values. In contrast, in panel (d), L_o allows it to take continuous values.

can be problematic because two very similar orientations can be treated as being different (if they occur in neighboring bins). Hence we propose the representation in panel (d) where the occlusion orientation variable is continuous. This pixel-based representation is well suited for deep networks using local and non-local cues and regression to estimate the continuous orientation variable.

Loss functions for occlusion relations. Given an image I we assign a pair of labels, $l = \{e, \theta\}$, to each pixel. Here $e \in \{1, 0\}$ is a binary indicator variable with $e = 1$ meaning that the pixel is located on a boundary. $\theta \in (-\pi, \pi]$ is an occlusion orientation variable defined at the boundaries, i.e. when $e = 1$, which specifies the tangent of the boundary and whose direction indicates border ownership using the “left” rule, see Fig. 3.2 (d) and Fig. 3.1 left. If $e = 0$, we set $\theta = NA$ and do not use these points for the occlusion loss computation.

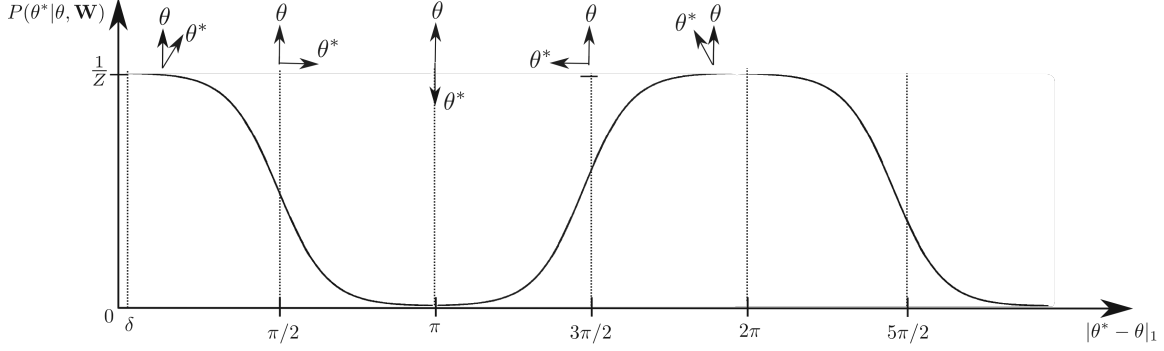


Figure 3.3: The orientation probability $P(\theta_j^*|\theta_j, \mathbf{W})$ as a function of the difference between the predicted and ground truth orientation, i.e. θ^* and θ in the figure.

For training, we denote the set of training data by $\mathcal{S} = \{(\mathbf{I}_i, \mathcal{L}_i)\}_{i=1}^N$, where N is the number of training images, and $\mathcal{L}_i = \{\mathbf{L}_{ei}, \mathbf{L}_{oi}\}$ are the ground truth annotations, where $\mathbf{L}_{ei}$ specifies the boundary and $\mathbf{L}_{oi}$ the occlusion orientation. Our goal is to design a DCNN that can learn a mapping function parameterized by $\mathbf{W}$, i.e. $f(\mathbf{I}_i : \mathbf{W})$, that can estimate the ground truth $\mathcal{L}_i$.

To learn the parameters $\mathbf{W}$, we define a loss function:

$$l_{doc}(\mathcal{S} : \mathbf{W}) = \frac{1}{N} \left(\sum_i l_e(\mathbf{I}_i, \mathbf{L}_{ei} : \mathbf{W}) + \sum_i l_o(\mathbf{I}_i, \mathbf{L}_{oi} : \mathbf{W}) \right) \quad (3.1)$$

where $l_e(\mathbf{I}, \mathbf{L}_e : \mathbf{W})$ is the loss for the boundaries, and $l_o(\mathbf{I}, \mathbf{L}_o : \mathbf{W})$ is the loss for the occlusion orientations. The boundary loss is the balanced sigmoid cross entropy loss, which is used by the state-of-the-art edge detector [XT15]. It is defined by:

$$l_e(\mathbf{I}, \mathbf{L}_e : \mathbf{W}) = -\beta \sum_{j:e_j=1} \log P(e_j^* = 1|\mathbf{I}, \mathbf{W}) - (1 - \beta) \sum_{j:e_j=0} \log P(e_j^* = 0|\mathbf{I}, \mathbf{W}) \quad (3.2)$$

where e_j and e_j^* are respectively the boundary ground truth and boundary prediction at pixel j , and $\beta = |\mathbf{L}_e^-|/|\mathbf{L}_e|$ is the percentage of pixels labelled as non-boundary for image $\mathbf{I}$. $P(e_j^* = 1|\mathbf{I}, \mathbf{W})$ is computed from the sigmoid output of the deep network at pixel j .

The occlusion orientation loss function strongly penalizes wrong directions (i.e. errors in border ownership using the “left” rule) but only weakly penalizes the tangent direction, as illustrated in Fig. 3.3. Let θ_j and θ_j^* respectively denote the occlusion orientation groundtruth and the estimation. Then the loss is:

$$\begin{aligned}
l_o(\mathbf{I}, \mathbf{L}_o : \mathbf{W}) &= - \sum_{j:e_j=1} \log P(\theta_j^* | \theta_j, \mathbf{W}) \\
\text{where, } P(\theta_j^* | \theta_j, \mathbf{W}) &= \frac{1}{Z} \begin{cases} 1 & : |\theta_j - \theta_j^*|_1 \in [0, \delta] \cup [2\pi - \delta, 2\pi + \delta] \\ \text{Sigmoid}(\alpha(f(|\theta_j - \theta_j^*|_1))) & : \text{otherwise} \end{cases} \\
f(|\theta_j - \theta_j^*|_1) &= \begin{cases} \pi/2 - |\theta_j - \theta_j^*|_1 & : |\theta_j - \theta_j^*|_1 \in [0, \pi] \\ |\theta_j - \theta_j^*|_1 - \pi & : |\theta_j - \theta_j^*|_1 \in (\pi, 2\pi] \\ 3\pi/2 - |\theta_j - \theta_j^*|_1 & : |\theta_j - \theta_j^*|_1 \in (2\pi, +\infty) \end{cases} \quad (3.3)
\end{aligned}$$

where $|x|_1$ is the absolute value of x . Z is the normalizing constant. This loss function has two hyper parameters α and δ , where α is a scale factor for the sigmoid function, which controls the strength at direction inverting points. δ controls a non-penalizing range when the θ_j^* is close enough to θ_j .

3.2.2 The network architecture

We present two DOC architectures, DOC-HED and DOC-DMLFOV, which are based respectively on the holistic-nested edge detector network (HED) [XT15] and the deeplab multi-scale large field of view DMLFOV network [CPK15]. We choose these networks because: (1) Both exploit local and non-local information and have multi scale outputs (important for occlusion). (2) Both were state-of-the-art on their assigned tasks (and remain highly competitive). HED for detecting edges in the BSDS dataset [AMF11], and DMLFOV for PASCAL semantic segmentation. Also they use different features, for edges or regions, which makes them interesting to compare.

Preliminaries. We briefly introduce the structures of HED and DMLFOV, see Fig. 3.4. HED is shown at the top left in Fig. 3.4. It is obtained by removing the fully connected layers of the VGG network [SZ14], enabling it to better capture low-level image details required for edge detection. The network produces side outputs at different levels of the network which are combined by a weighted fusion, yielding multi-scale outputs. DMLFOV is shown at the top right of Fig. 3.4. This network contains two fully connected (fc) layers which has a much smaller parameter space (1024 dimension) comparing to the original fc layers (4096 dimension) in the VGG network. This network also produces side outputs which are combined for the final output.

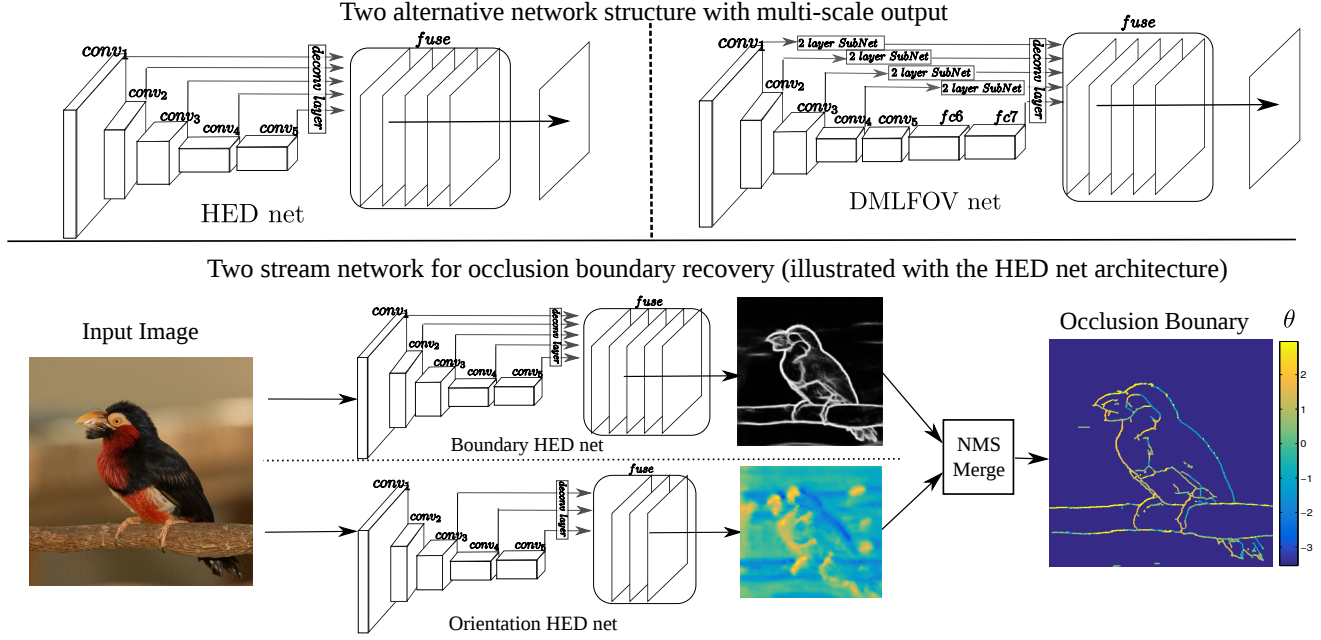


Figure 3.4: Top: We explored two alternative DOC architectures for occlusion recovery: (i) HED, and (ii) DMLFOV. Bottom: for inference, we first apply a two stream network (shown for HED) to predict pixel-wise boundaries and the occlusion orientations respectively. Then, we apply non-maximum suppression (NMS) to the boundaries, merge the two predictions, and recover the occlusion boundaries.

Two streams and up sampling. To adapt HED and DMLFOV to estimate occlusion relations we modify them in two ways: (1) For pixel-based tasks, requiring precise localization of boundaries and estimation of occlusion orientation, we need to up sample the network outputs, to correct for low-resolution caused by max pooling (particularly important for DMLFOV which addressed the less precise task of semantic segmentation). To achieve this we combine the “hole” algorithm [CML14] with deconvolution up-sampling [LSD15b]. (2) To adapt HED and DMLFOV to work on the occlusion representation, see previous section, we adopt a two stream network (encouraged by prior work [WSL15b] when using deep networks to address two tasks simultaneously). For estimating the boundaries we keep the original network structure. For estimating the occlusion orientation, which requires a large range of context, we combine outputs only at higher levels of the network (experiments shown that low-level outputs were too noisy to be useful). Thus, for the DOC-HED network, we drop the side output predictions before “conv3” (as illustrated at bottom of Fig. 3.4),

and for DOC-DMLFOV we drop the predictions (also from side outputs) before “conv4”.

Training phase. We train DOC-HED and DOC-DMLFOV using the pixel-based representations described in the previous section. They are trained on both the BSDS border ownership dataset [RFM06] and on a new dataset, based on PASCAL VOC, which we will describe in the next section.

Testing phase. Given an input image, DOC outputs a boundary map and an occlusion orientation map (from the two streams). To combine the results, we first perform non-maximum suppression (NMS) on the boundary map, using the method as [DZ15]. Then we obtain the occlusion orientation for each edge pixel (i.e. pixel that we have classified as boundary) from the orientation map. Finally we adjust the orientation estimation to ensure that neighboring pixels on the curve have similar orientations. More specifically, we align the orientation to the tangent line estimated from the boundary map since we trust the accuracy of the predicted boundaries. Formally, at a pixel j , the predicted orientation and one direction of the tangent line are θ_j and θ_{tj} respectively. We set θ_j to be θ_{tj} if $|\theta_j - \theta_{tj}| \bmod 2\pi \in [0, \pi/2) \cup (3\pi/2, 2\pi]$, and to the reverse direction of θ_{tj} otherwise. Finally, motivated by the observation that the results are more reliable if the boundary and orientation predictions are consistent, we take $c_{oj} = |\cos(|\theta_j - \theta_{tj}|)|_1$ as the confidence score for the occlusion orientation prediction at pixel j . Finally, given the predicted confidence score c_{ej} from the boundary network outputs, our final confidence score for the occlusion boundary at pixel j is defined to be $c_{ej} + c_{oj}$.

3.3 Pascal instance occlusion dataset (PIOD)

A large dataset is of critical for training and evaluating deep network models. The BSDS border ownership dataset [RFM06] helped pioneer the study of occlusion relations on natural images but is limited because it only contains 200 images, and hence it may not be able to capture the range of occlusion relations that happen in natural images (our experiments will address how well models trained on one dataset transfer to another).

We choose to annotate occlusion on the PASCAL VOC dataset because it is large, contains well-selected images, and other researchers have already annotated the boundaries for 20 object

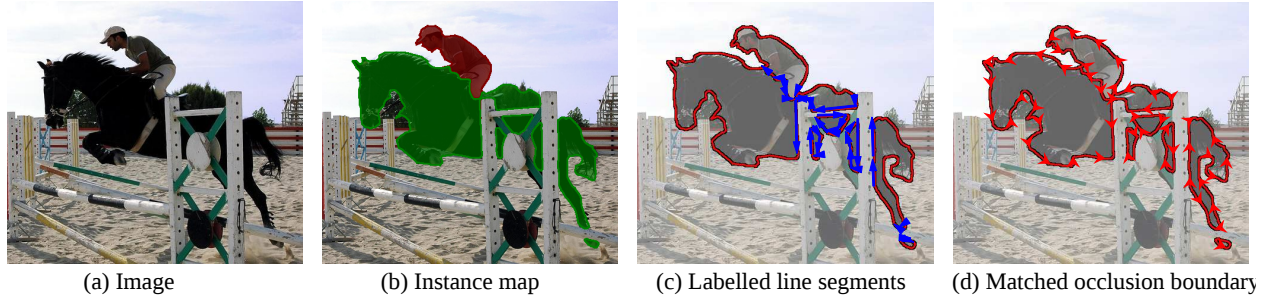


Figure 3.5: The annotation process of our PIOD. Given an image, we provide two annotated maps, i.e. (b) the semantic instance map and (c) the generated boundary map. An annotator needs to supplement the boundary map with directed line segments following the “left” rule. We assume the objects occlude background by default, so the annotator only needs to label the boundaries violating this rule or between adjacent instances. Finally, we match the labelled line segments to all the boundaries as shown in (d).

instances [HAB11, CML14]. These object boundary annotations are very reliable because the annotators were given clear instructions and consistency checks were performed. Hence our annotation task reduces to annotating border ownership by specifying the directions of the occlusion orientation. Our strategy is to annotate the directions of line segments to specify occlusion orientations, or boundary ownership, using the “left” rule. We do this by a two stage process, as shown in Fig. 3.5. The annotators are asked to label directed straight line segments which lie close to the object boundaries and whose directions specify the border ownership. The second stage is performed by an algorithm which matches the directed line segments to the annotated boundaries. The idea is that the first stage can be done quickly, since the line segments do not have to lie precisely on the edges, while the second stage gives an automated way to exploit the existing boundary annotations [HAB11, CML14].

Stage 1: Annotate with directed line segments For each image, the annotator is given two annotation maps: (i) the boundary map, and (ii) the semantic instance map [HAB11, CML14]. We assume the object is occluding the background, so we only annotate the boundaries between any two adjacent object instances and the boundaries where objects are occluded by background. For each boundary segment, the annotator draw a directed line segment close to the boundary whose direction indicates the occlusion orientation based on the “left” rule.

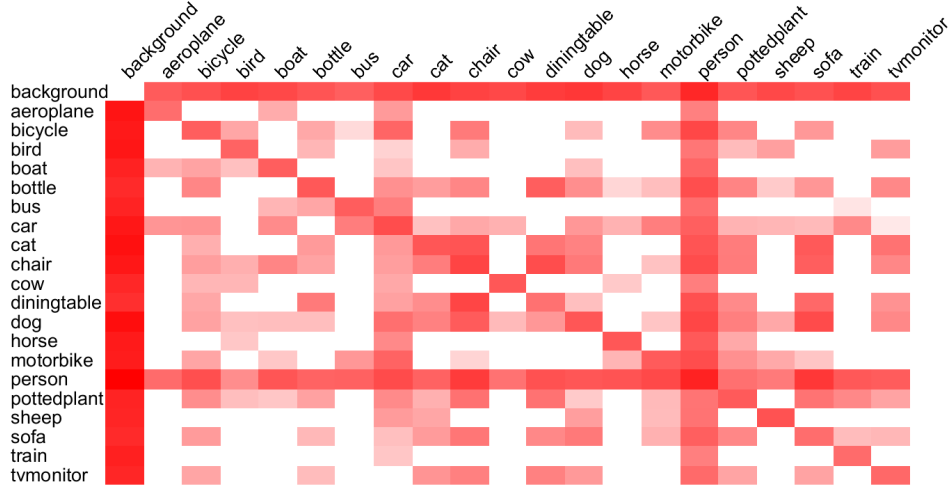


Figure 3.6: The occlusion relationship matrix shows the frequency of occlusion between different classes (red means high). The horizontal axis denotes the occluded objects.

Stage 2: Matching directed line segments to object boundaries. To associate the directed line segments to the boundary map, we developed a matching tool which maps the annotated line segments to the boundaries of all object instances. Our ground truth occlusion boundaries are then represented by a set of boundary fragments, similar to [HEH11]. Each fragment is associated with a start and end point of a directed line segment. Finally, we convert this representation to an occlusion orientation map where each pixel on the object boundary is assigned an occlusion orientation value indicating the local occlusion direction. This process is shown in Fig. 3.5, where we give images with our labelled results overlaid.

Statistics of Occlusion Relations In Fig. 3.6, we show the frequency statistics of the occlusion relationships between different classes of objects. Each row indicates how frequently an object of a particular class occludes other classes of objects (or background). Note that a large number of occlusions are due to objects occluding the background. Observe also that "persons" appear very frequently in the table because humans interact with many other object classes. These occlusion statistics are useful to understand the biases in our PIOD dataset.

3.4 Experiments

We experimented with our DOC approach on the BSDS ownership dataset [RFM06] and our new PASCAL instance occlusion dataset (PIOD). As mentioned before, these datasets differ by size (PIOD is two orders of magnitude bigger) and boundary annotations (PIOD contains only the boundaries of the 20 PASCAL objects while BSDS includes internal and background edges).

In this section, we first propose a more reliable criteria for occlusion boundary evaluation than that used by [RFM06, TFA15], which was also questioned by previous work [LL09] (see Sec. 3.4.1). Then, we conduct extensive experiments with the DOC networks as described in Sec. 3.2.2. These show that DOC significantly outperforms the state-of-the-art [TFA15]. Both DOC-HED and DOC-DMLFOV perform well, so we perform experiments on both PIOD and on the BSDS ownership data to gain insights about the network architectures for future research. We also study transfer between the two datasets, and other issues.

Implementation details For the orientation loss function in Eqn. (3.3), we set $\alpha = 4$ and $\delta = 0.05$ respectively, chosen using the validation set. For learning both networks, DOC-HED and DOC-DMLFOV, we used the deep supervision strategy [XT15], with the learning rate and stage-wise training the same as for HED and DMLFOV respectively. We initialized the models using versions of HED and DMLFOV released by the authors.

For learning on the BSDS ownership dataset, we followed the HED strategy and use adaptive input size for training and testing by setting the “batchsize” to 1 and “itersize” to 10. When learning on PIOD, since the number of images is very large, to save training time, we resize all the input images to 386×386 by keeping the aspect ratio and padding with zeros. We set the “batchsize” to 15 and “itersize” to 2. For both datasets, we augment each image as proposed by HED. We implement all our models based on the published paraset [LRB15] fork of Caffe [JSD14], which includes both the “hole” algorithm and deconvolution. We also merge the implemented input and cross entropy loss layers from the code released by HED.

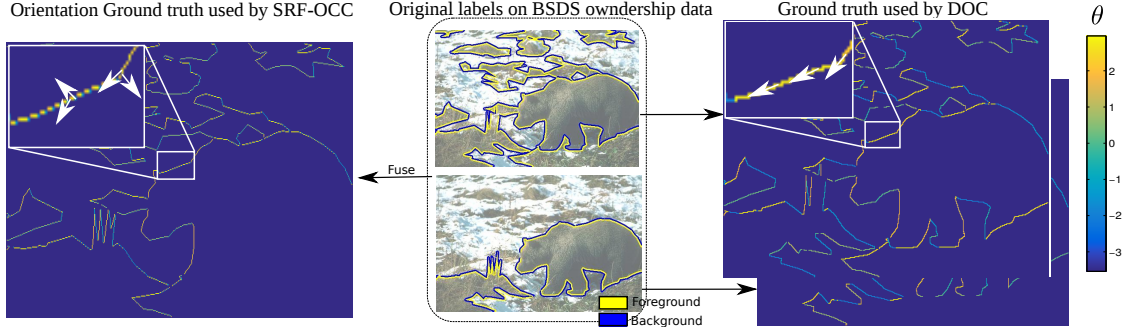


Figure 3.7: Center: the two ground truth maps for each image in the BSDS ownership data. Left: limitation of the orientation map generated by SRF-OCC [TFA15] for occlusion evaluation. In the white rectangle, the white arrows show the quantized ground truth orientation at corresponding pixels, which is not smooth or intuitively correct. Also, at bottom right, label inconsistent ground truth edges are discarded when fusing multiple maps. Right: our proposed multiple ground truth occlusion orientation maps for learning and evaluation.

3.4.1 Evaluation criteria

Specifying a criterion for evaluating occlusion relations is not easy. The problem is that it involves two tasks: detecting boundaries and specifying border ownership. One proposed criteria [RFM06] computes the percentage of the pixels for which the occlusion relations are estimated correctly. But this criteria was criticized [LL09] because it depends on the selected pixel matching method (between the estimates and the groundtruth boundaries) and the choice of threshold for the edge detector. e.g., a high threshold for the edge detector will detect fewer boundaries but may label their border ownership more accurately. Another criteria was proposed by [TFA15], who released evaluation code. But, see Fig. 3.7, we found two problems that may lead to unreliable results. The first is that they quantize the occlusion orientation angle to take 8 values which can lead to errors, see the white rectangle on the left of Fig. 3.7. This quantization problem is enhanced because the orientation was computed based on a local pixel-wise gradient (relying on a pair of neighbored pixels with 8 connections). The second problem is they evaluate on the BSDS ownership dataset which combines boundary maps from different annotators but without checking for consistency [HYK13], which may bias the evaluation since the error cases due to label inconsistency are dropped.

To address these two problems, we first propose to compute the orientation based on a local boundary fragment of length 10 pixels, as used by [HEH11], yielding a smoother and intuitively more reasonable ground truth orientation for evaluation, see right of Fig. 3.7. Secondly, for evaluating the occlusion relations, we propose a new criteria called the *Occlusion accuracy w.r.t. boundary recall Curve*, which we refer to as the *AOR curve*. This adapts an idea from the standard PR curve used to evaluate edge detection.

Formally, given the occlusion boundary estimation result with threshold t , we find the correctly detected boundary pixels and their corresponding ground truth pixels by matching them to a ground truth map by the standard edge correspondence method [AMF11]¹. Then for each pixel i on the estimated boundaries, its predicted occlusion orientation θ_i^* is compared to the corresponding ground truth orientation θ_i . We keep the match if $|\theta_i - \theta_i^*| \in [0, \pi/2) \cup (3\pi/2, 2\pi]$, but drop it as a false positive otherwise. After matching all the pixels we obtain two values: (i) the recall rate $R_e(t)$ of the ground truth boundary, and (ii) the accuracy $A_o(t)$ of occlusion orientation prediction given the recalled boundaries.

By varying the threshold t , we can summarize the relationship between $R_e(t)$ and $A_o(t)$ by a curve comparing the accuracy of border ownership as a function of the amount of boundary recalled (i.e. each point on the curve corresponds to a value of the threshold t). In our experiments, we draw the curves to uniformly sample 33 thresholds. For the AOR curve, the accuracy at high recall is most important since more test data used for evaluation yields more reliable indication for the model’s ability. We will release our developed evaluation code and ground truth for reproducing all our results.

3.4.2 Performance comparisons.

We extensively compare our deep occlusion (DOC) approaches with different settings and configurations of the HED [XT15] and DMLFOV [CPK15] networks. We also compare DOC-HED and DOC-DMLFOV to the state-of-the-art occlusion recovering algorithm [TFA15] which we refer to as SRF-OCC (it uses structured random forests). In Fig. 3.8, we see almost all our models outperform

¹We use the toolbox from the BSDS benchmark website.

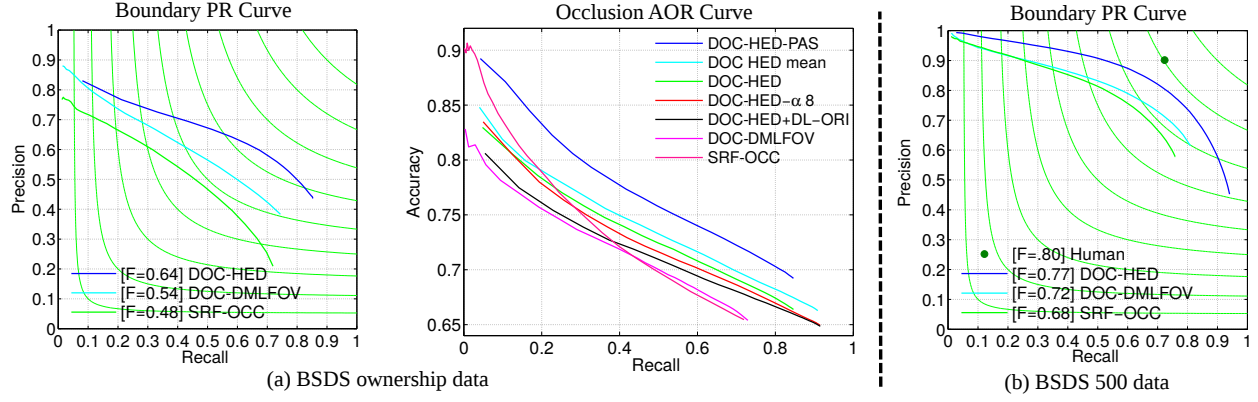


Figure 3.8: Quantitative comparison on BSDS ownership data. SRF-OCC [TFA15] is the baseline model. In (b), we show the edge detection performance on BSDS 500 testing data with models trained from the BSDS ownership data (with only 100 images). This shows the DOC-HED model we trained are comparable to those in the HED paper (best viewed in color). Details are in Sec. 3.4.2.

SRF over both datasets over 6%, showing the effectiveness of our approach.

BSDS ownership data. The BSDS ownership dataset contains 100 training images and 100 testing images. We evaluate our deep networks on this dataset although its small size makes them challenging to train. The edge detection comparisons, see left of Fig. 3.8(a), show that DOC-HED performs best, DOC-DMLFOV is the runner up and SRF-OCC performs less well.

Observe that the results for DOC-HED are not as good as that reported for HED by [XT15] when trained and tested on the full BSDS dataset. So we evaluated our trained DOC-HED model over the standard BSDS 200 test images and give the results in Fig. 3.8(c), showing performance very similar to HED (Fusion-output). We think the difference is due to three reasons. Firstly, in order to give fair comparisons to SRF-OCC we train on 100 image only (unlike 300 for HED). Secondly, the images in BSDS ownership data are a non-randomly selected subset of the full BSDS dataset, where the images were chosen to study occlusion and edges inside are harder to detect. Thirdly, each image in this data only uses two ground truth annotations which might introduce labeling noise [HYK13].

On the right of Fig. 3.8(a), we give results for occlusion relations using our AOR curve. Trained on just 100 images, and tested with single scale image input, the DOC-HED network (green line)

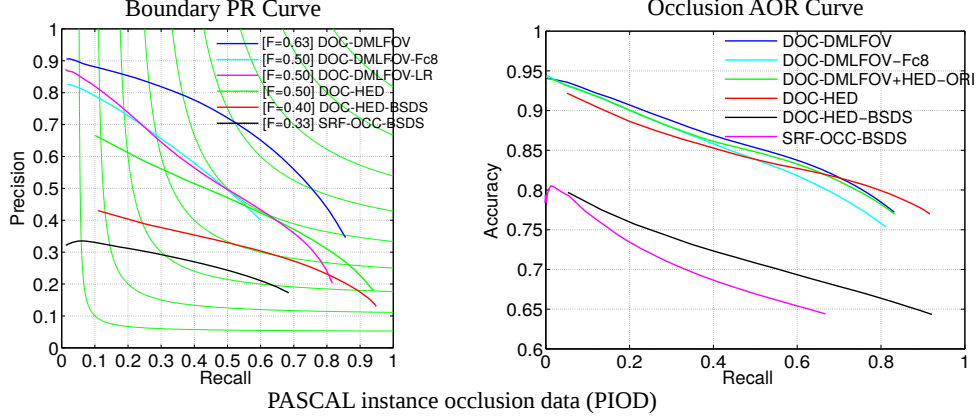


Figure 3.9: Quantitative comparison over our PIOD data.

performs best, outperforming SRF-OCC when the edge recall rate is higher than 0.3, and the margin goes above 4% at high recall rate of 0.7. The relatively weak performance of the DOC-DMLFOV network (pink line) is probably because it is a more complex network than HED and does not have enough data in BSDS ownership to train it properly. Its performance is lower than DOC-HED network, but is still competitive with SRF-OCC for recall above 0.7. Finally, we investigate transfer by pre-training DOC-HED-PAS (blue line) on PIOD and then fine-tuning it on BSDS ownership data. This improves performance by another 3%, yielding an average improvement of 6% over the SRF-OCC model on the BSDS ownership dataset. This illustrates the advantages of having more data when training deep networks, as well as the ability to transfer models trained on PASCAL to BSDS. Finally, we give visualization results in Fig. 3.10(a), illustrating that our DOC model recovers better semantic boundaries.

PASCAL instance occlusion dataset (PIOD). PIOD contains 10,100 images, and we take 925 images from the VOC 2012 validation set for testing. We show performance for semantic edge detection at the left of Fig. 3.9. Note there is a difference with BSDS which includes many low-level edges, while PIOD contains only object boundaries. The figure shows that DOC DMLFOV provides the best performance, presumably because it captures strong long-range context, while DOC-HED performs comparatively weaker in this case. In addition, we study transfer from BSDS ownership to PIOD and show that DOC-HED-BSDS (i.e. trained on BSDS) outperforms SRF-OCC-BSDS, but both perform much worse than the deep networks trained on PIOD.

For estimating occlusion relations, see right of Fig. 3.9, DOC-DMLFOV performs best, but only a little better than DOC-HED (i.e. by around 1.5%) and worse than DOC-HED for recall higher than 0.78. This is because, for the boundaries which are correctly estimated, DOC-HED also gives accurate occlusion orientation estimates.

We also evaluated the ability of SRF-OCC and DOC-HED models when trained only on BSDS. As shown in the figure, DOC-HED-BSDS outperforms SRF-OCC-BSDS significantly on PIOD by a margin of 5% and is higher at every level of recall, showing better ability of deep networks despite the small amount of training data. Some examples visualizing our results are shown in Fig. 3.10(b). Notice that many of the false positives in the DOC predictions are intuitively correct but were not labelled. The deep networks trained on PIOD data do much better than those trained on BSDS.

Additional comparisons on the two datasets. We performed additional experiments to get more insight about the abilities of the different models.

Tuning of α . Recall that α is the parameter controlling the sharpness of the occlusion orientation term in Eqn. (3.3). As shown in the right of Fig. 3.8(a) (DOC-HED- α 8: red curve), if we set α to 8 then performance drops slightly because it only weakly penalizes the closeness between θ and θ^* , see Fig. 3.3. We found the optimal value to be 4, using a validation set, and fixed this in the experiments.

Scales of input images. Multi-scale training and testing usually gives better results. On the right of Fig. 3.8(a) (DOC-HED mean: cyan curve), we show the results from averaging three images scale ([0.5, 1.0, 1.5]) outputs from the DOC-HED network. But multi-scale only gave marginal improvement. This suggests that for boundary detection, multi-scale networks and multi-scale input contain similar information.

Multi-scales network vs. Single scale network. We compared the final fusion output (DOC DMLFOV) vs. single side output (from the “fc8” layer) based on the DOC DMLFOV network over PIOD. As shown on the left of Fig. 3.9(b) (DOC DMLFOV-Fc8: cyan curve), single side output gives much weaker performance for boundary detection since it localizes the edges worse compared to multi-scale. On the right of Fig. 3.9(b), DOC DMLFOV-Fc8 performs well but is still weaker than DOC DMLFOV for occlusion recovery. This shows, as expected, that high level features

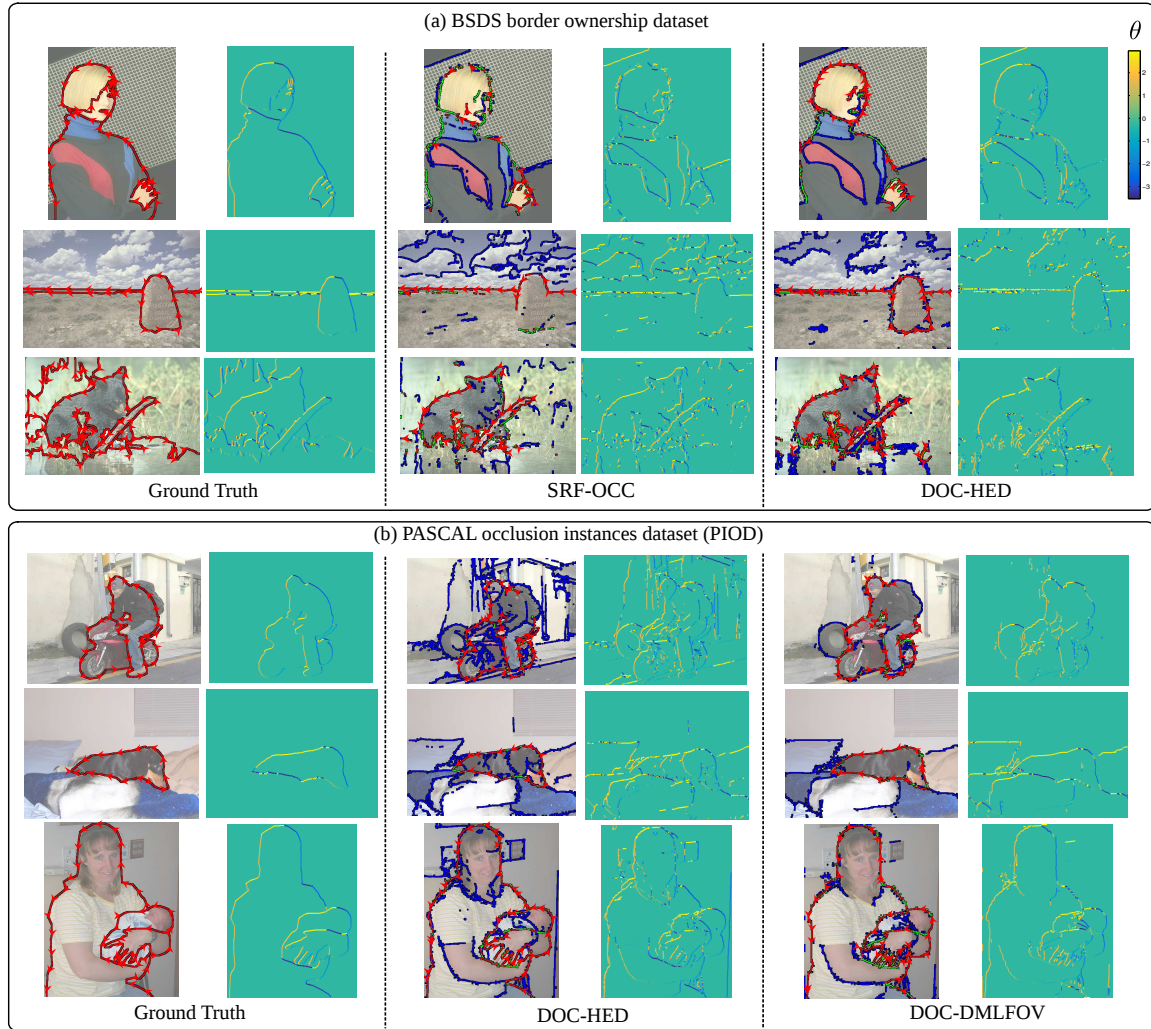


Figure 3.10: Qualitative comparisons (best viewed in color). At the left side of each column, we show algorithm results compared with ground truth. The “red” pixels with arrows are correctly labelled occlusion boundaries, the “green” pixels are correctly labeled boundaries but incorrect occlusion, and the “blue” pixels are false positive boundaries. At the right of each column, we show the occlusion boundaries by a 2.1D relief sculpture. In the figure, the foreground regions are raised (embossed). (a) Comparisons on the BSDS ownership data between SRF-OCC [TFA15] and DOC-HED. (b) Comparisons on PIOD between DOC-HED and DOC-DMLFOV. Note for some images, some internal occlusion boundaries are recovered (although they are not labelled correct), e.g., the tire on the bike and the woman’s right arm. This that DOC can generalize from boundaries to some internal edges. We give more examples in Fig.1 of the supplementary material.

contribute most to the occlusion orientation estimation.

High resolution loss vs. Low resolution loss. Unlike the original loss based on down-sampled ground truth used by DMLFOV for training semantic segmentation [CPK15], our loss is computed from the label map at the original image resolution. As shown when compared, see left of Fig. 3.9(b), with the low resolution model (DMLFOV-LR: pink curve), this strategy improves both boundary detection and occlusion orientation.

Replacing the boundary detector network stream. Because the AOR curve performs a joint evaluation of boundary detection and border ownership, we must see how DOC-HED and DOC DMLFOV perform on each individual task. We already compared them for boundary detection, so we now replace the DOC boundary network to see how the performance of occlusion estimation depends on the two alternative architectures.

In Fig. 3.8(a) (DOC-HED+DL-ORI: black curve), we use DOC-HED for boundary detection but DOC-DMLFOV for estimating the occlusion orientation. This gives a performance drop of 2% compared to DOC-HED. This shows, for dataset with internal edges like BSDS, DOC-HED also outperforms DOC-DMLFOV on occlusion prediction. In Fig. 3.9(b) (DOC-DMLFOV+HED-ORI: green curve), we apply the same strategy but use DOC-DMLFOV for boundaries and DOC-HED for occlusion orientation, giving a result close to that from DOC-DMLFOV. This shows when training on the large dataset PIOD, DOC-HED (smaller network) can perform as well as DOC-DMLFOV for occlusion estimation. These experiments show DOC-HED performs well in general for occlusion estimation.

3.5 Conclusion and future work

In this paper, we designed an end-to-end deep occlusion network (DOC) for estimating occlusion relations. We gave two variants, DOC-HED and DOC-DMLFOV, and show that they both give big improvements over state-of-the-art methods. We also constructed a new dataset PIOD for studying occlusion relations which is two orders of magnitude larger than comparable datasets. We show that PIOD enables better training and testing of deep networks for estimating occlusion relations. We also show good transfer from PIOD to the smaller BSDS border ownership dataset, but that methods

trained on BSDS border ownership are sub-optimal on PIOD. Our results show that DOC-HED and DOC-DMLFOV have complementary strengths which can be combined in future work. We hope that our PIOD dataset will serve as a resource to stimulate research in this important research area.

3.6 Appendix

In this appendix, we show additional qualitative comparison results from both the BSDS ownership dataset [RFM06] and our PIOD dataset in Fig. 3.12. Notice that in PIOD, many of the false positives from our predictions are intuitively actually correct but were not labelled.

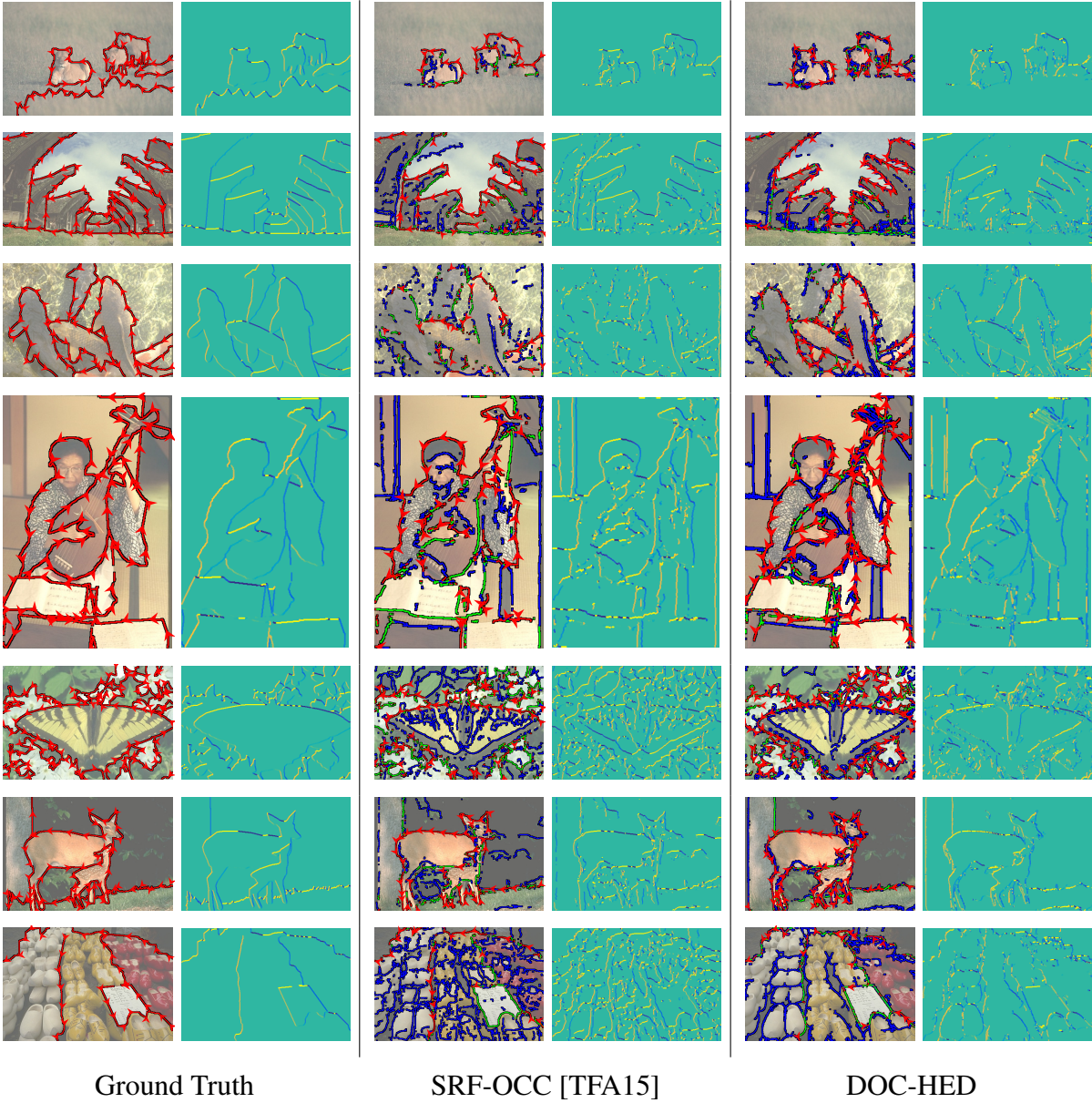


Figure 3.11: Qualitative comparison examples over the BSDS border ownership data (Best view in color).



Figure 3.12: Qualitative comparison examples over the PIOD (Best view in color).

CHAPTER 4

Joint segmentation and 3D geometry estimation from a single image

Recent efforts to estimate the 2.5D layout of a depicted scene from a single image, such as per-pixel depths and surface normals, has yielded high-quality outputs respecting both the global scene layout and fine object detail [BRG16, EPF14, EF15, WFG15]. Upon closer inspection, however, the predicted depths and normals may fail to be consistent with the underlying surface geometry. For example, consider the depth and normal predictions from the contemporary approach of Eigen and Fergus [EF15] shown in Figure 1. Notice the significant distortion in the predicted depth corresponding to the depicted planar surfaces, such as the back wall and cabinet. We argue that such distortion arises from the fact that the 2.5D predictions (i) are made independently per pixel from appearance information alone, and (ii) do not explicitly take into account the underlying surface geometry. When 3D geometry has been used, e.g., [WFG15], it often consists of a boxy room layout constraint, which may be too coarse and fail to account for local planar regions that do not adhere to the box constraint. Moreover, when multiple 2.5D predictions are made (e.g., depth and normals), they are not explicitly enforced to agree with each other.

To overcome the above issues, we introduce an approach to identify depicted 3D planar regions in the image along with their spatial extent, and to leverage such planar regions to regularize the depth and surface normal outputs. We formulate our approach as a four-stream convolutional neural network (CNN), followed by a dense conditional random field (DCRF). The four-stream CNN independently predicts at each pixel the surface normal, depth, and likelihoods of planar region and planar boundary. The four cues are integrated into a DCRF, which encourages the output depths and normals to align with the inferred 3D planar surfaces while maintaining fine detail within objects. Furthermore, the output depths and normals are explicitly encouraged to agree with each other.

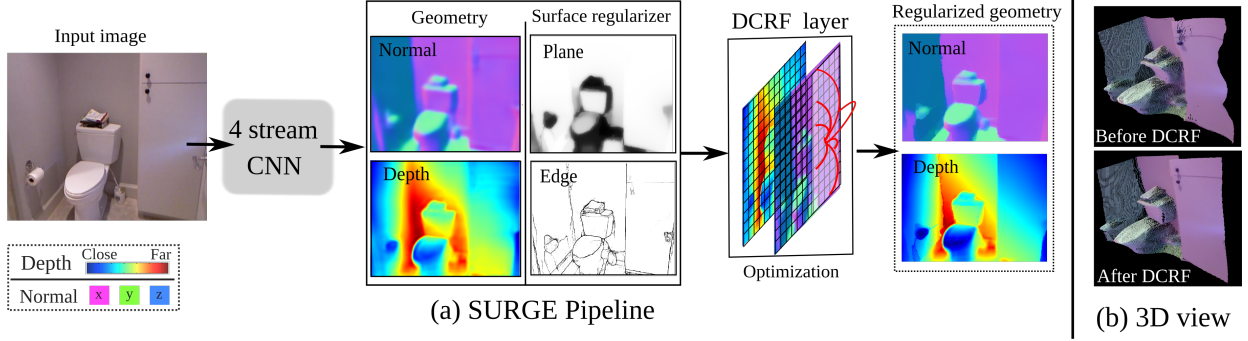


Figure 4.1: Framework of SURGE system. (a) We induce surface regularization in geometry estimation through DCRF, and enable joint learning with CNN, which largely improves the visual quality (b).

We show that our DCRF is differentiable with respect to depth and surface normals, and allows back-propagation to the depth and normal CNNs during training. We demonstrate that the proposed approach shows relative improvement over the base CNNs for both depth and surface normal prediction on the NYU v2 dataset using the standard evaluation criteria, and is significantly better when evaluated using our proposed plane-wise criteria.

4.1 Related work

From a single image, traditional geometry estimation approaches rely on extracting visual primitives such as vanishing points and lines [HEH07] or abstract the scenes with major plane and box representations [SFP13, SSP14]. Those methods can only obtain sparse geometry representations, and some of them require certain assumptions (*e.g.* Manhattan world).

With the advance of deep neural networks and their strong feature representation, dense geometry, i.e., pixel-wise depth and normal maps, can be readily estimated from a single image [EPF14]. Long-range context and semantic cues are also incorporated in later works to refine the dense prediction by combining the networks with conditional random fields (CRF) [WSL15a, LSL15a, LSD15a, WSL15a]. Most recently, Eigen *et al.* [EF15] further integrate depth and normal estimation into a large multi-scale network structure, which significantly improves the geometry estimation accuracy. Nevertheless, the output of the networks still lacks regularization over planar surfaces due to the

adoption of pixel-wise loss functions during network training, resulting in unsatisfactory experience in 3D image editing applications.

For inducing non-local regularization, DCRF has been commonly used in various computer vision problems such as semantic segmentation [CPK15, ZJR15], optical flow [KK12b] and stereo [BAS15]. However, the features for the affinity term are mostly simple ones such as color and location. In contrast, we have designed a unique planar surface affinity term and a novel compatibility term to enable 3D planar regularization over geometry estimation.

Finally, there is also a rich literature in 3D reconstruction from RGBD images [IYF15, SHK12, XF12, GH13, SLX15], where planar surfaces are usually inferred. However, they all assume that the depth data have been acquired. To the best of our knowledge, we are the first to explore using planar surface information to regularize dense geometry estimation by only using the information of a single RGB image.

4.2 Overview

Fig. 5.2 illustrates our approach. An input image is passed through a four-stream convolutional neural network (CNN) that predicts at each pixel a surface normal, depth value, and whether the pixel belongs to a planar surface or edge (i.e., edge separating different planar surfaces or semantic regions), along with their prediction confidences. We build on existing CNNs [EF15, XT15] to produce the four maps.

While the CNNs for surface normals and depths produce high-fidelity outputs, they do not explicitly enforce their predictions to agree with depicted planar regions. To address this, we propose a fully-connected dense conditional random field (DCRF) that reasons over the CNN outputs to regularize the surface normals and depths. The DCRF jointly aligns the surface normals and depths to individual planar surfaces derived from the edge and planar surface maps, all while preserving fine detail within objects. Our DCRF leverages the advantages of previous fully-connected CRFs [KK12a] in terms of both its non-local connectivity, which allows propagation of information across an entire planar surface, and efficiency during inference. We present our DCRF formulation in Section 4.3, followed by our algorithm for joint learning and inference within a CNN

in Section 4.4.

4.3 DCRF for Surface Regularized Geometry Estimation

In this section, we present our DCRF that incorporates plane and edge predictions for depth and surface normal regularization. Specifically, the field of variables we optimize are depths, $\mathbf{D} = \{d_i\}_{i=1}^K$, where K is number of the pixels, and normals, $\mathbf{N} = \{\mathbf{n}_i\}_{i=1}^K$, where $\mathbf{n}_i = [n_{ix}, n_{iy}, n_{iz}]^T$ indicates the 3D normal direction.

In addition, as stated in the overview (Sec. 4.2), we have four types of information from the CNN predictions, namely a predicted normal map $\mathbf{N}_o = \{\mathbf{n}_i^o\}_{i=1}^K$, a depth map $\mathbf{D}_o = \{d_i^o\}_{i=1}^K$, a plane probability map $\mathbf{P}_o$ and edge predictions $\mathbf{E}_o$. Following the general form of DCRF [KK12b], our problem can be formulated as,

$$\min_{\mathbf{N}, \mathbf{D}} \sum_i \psi_u(\mathbf{n}_i, d_i | \mathbf{N}_o, \mathbf{D}_o) + \lambda \sum_{i,j, i \neq j} \psi_r(\mathbf{n}_i, \mathbf{n}_j, d_i, d_j | \mathbf{P}_o, \mathbf{E}_o) \quad \text{with } \|\mathbf{n}_i\|_2 = 1 \quad (4.1)$$

where $\psi_u(\cdot)$ is a unary term encouraging the optimized surface normals $\mathbf{n}_i$ and depths d_i close to the output $\mathbf{n}_i^o$ and d_i^o from the networks. $\psi_r(\cdot, \cdot)$ is a pairwise fully connected regularization term depending on the information from the plane map $\mathbf{P}_o$ and edge map $\mathbf{E}_o$, where we seek to encourage consistency of surface normals and depths within planar regions with the underlying depicted 3D planar surfaces. Also, we constrain the normal predictions to have unit length. Specifically, the definition of unary and pairwise in our model are presented as follows.

4.3.1 Unary terms

Motivated by Monte Carlo dropout [SHK14], we notice that when forward propagating multiple times with dropout, the CNN predictions have different variations across different pixels, indicating the prediction uncertainty. Based on the prediction variance from the normal and depth networks, we are able to obtain pixel-wise confidence values w_i^n and w_i^d for normal and depth predictions. We leverage such information to DCRF inference by trusting the predictions with higher confidence while regularizing more over ones with low confidence. By integrating the confidence values, our

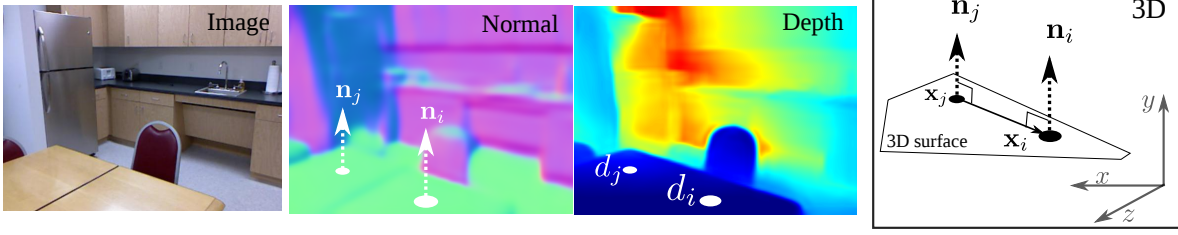


Figure 4.2: The orthogonal compatibility constraint inside the DCRF. We recover 3d points from the depth map and require the difference vector to be perpendicular to the normal predictions.

unary term is defined as,

$$\psi_u(\mathbf{n}_i, d_i | \mathbf{N}_o, \mathbf{D}_o) = \frac{1}{2} w_i^n \psi_n(\mathbf{n}_i | \mathbf{n}_o) + \frac{1}{2} w_i^d \psi_d(d_i | d_o), \quad (4.2)$$

where $\psi_n(\mathbf{n}_i | \mathbf{n}_o) = 1 - \mathbf{n}_i \cdot \mathbf{n}_o$ is the cosine distance between the input and output surface normals, and $\psi_d(d_i | d_o) = (d_i - d_o)^2$ is the squared difference between input and output depth.

4.3.2 Pairwise term for regularization.

We follow the convention of DCRF with Gibbs energy [KK13] for pairwise designing, but also bring in the confidence value of each pixel as described in Sec. 4.3.1. Formally, it is defined as,

$$\begin{aligned} \psi_r(\mathbf{n}_i, \mathbf{n}_j, d_i, d_j | \mathbf{P}_o, \mathbf{E}_o) &= (w_{i,j}^n \mu_n(\mathbf{n}_i, \mathbf{n}_j) + w_{i,j}^d \mu_d(d_i, d_j, \mathbf{n}_i, \mathbf{n}_j)) A_{i,j}(\mathbf{P}_o, \mathbf{E}_o), \\ \text{where, } w_{i,j}^n &= \frac{1}{2}(w_i^n + w_j^n), w_{i,j}^d = \frac{1}{2}(w_i^d + w_j^d) \end{aligned} \quad (4.3)$$

Here, $A_{i,j}$ is a *pairwise planar affinity* indicating whether pixel locations i and j belong to the same planar surface derived from the inferred edge and planar surface maps. $\mu_n()$ and $\mu_d()$ regularize the output surface normals and depths to be aligned inside the underlying 3D plane. Here, we use simplified notations, i.e. $A_{i,j}$, $\mu_n()$ and $\mu_d()$ for the corresponding terms.

For the compatibility $\mu_n()$ of surface normals, we use the same function as $\psi_n()$ in Eqn. (4.2), which measures the cosine distance between $\mathbf{n}_i$ and $\mathbf{n}_j$. For depths, we design an *orthogonal compatibility* function $\mu_d()$ which encourages the normals and depths of each adjacent pixel pair to be consistent and aligned within a 3D planar surface. Next we define $\mu_d()$ and $A_{i,j}$.

Orthogonal compatibility: In principle, when two pixels fall in the same plane, the vector connecting their corresponding 3D world coordinates should be perpendicular to their normal directions, as illustrated in Fig. 4.2. Formally, this orthogonality constraint can be formulated as,

$$\mu_d(d_i, d_j, \mathbf{n}_i, \mathbf{n}_j) = \frac{1}{2} (\mathbf{n}_i \cdot (\mathbf{x}_i - \mathbf{x}_j))^2 + \frac{1}{2} (\mathbf{n}_j \cdot (\mathbf{x}_i - \mathbf{x}_j))^2, \text{ with } \mathbf{x}_i = d_i \mathbf{K}^{-1} \mathbf{p}_i. \quad (4.4)$$

Here $\mathbf{x}_i$ is the 3D world coordinate back projected by 2D pixel coordinate $\mathbf{p}_i$ (written in homogeneous coordinates), given the camera calibration matrix $\mathbf{K}$ and depth value d_i . This compatibility encourages consistency between depth and normals.

Pairwise planar affinity: As noted in Eqn. (4.3), the planar affinity is used to determine whether pixels i and j belong to the same planar surface from the information of plane and edge. Here $\mathbf{P}_o$ helps to check whether two pixels are both inside planar regions, and $\mathbf{E}_o$ helps to determine whether the two pixels are belong to the same planar surface. Here, for efficiency, we chose the form of Gaussian bilateral affinity to represent such information since it has been successfully adopted by many previous works with efficient inference, e.g. in discrete label space for semantic segmentation [?] or in continuous label space for edge-awared smoothing [BAS15, KK12b]. Specifically, following the form of bilateral filters, our planar surface affinity is defined as,

$$\mathbf{A}_{i,j}(\mathbf{P}_o, \mathbf{E}_o) = p_i p_j (\omega_1 \kappa(\mathbf{f}_i, \mathbf{f}_j; \theta_\alpha) \kappa(\mathbf{c}_i, \mathbf{c}_j; \theta_\beta) + \omega_2 \kappa(\mathbf{c}_i, \mathbf{c}_j; \theta_\gamma)), \quad (4.5)$$

where $\kappa(\mathbf{z}_i, \mathbf{z}_j; \theta) = \exp(-\frac{1}{2\theta^2} \|\mathbf{z}_i - \mathbf{z}_j\|^2)$ is a Gaussian RBF kernel. p_i is the predicted value from the planar map $\mathbf{P}_o$ at pixel i . $p_i p_j$ indicates that the regularization is activated when both i, j are inside planar regions with high probability. $\mathbf{f}_i$ is the appearance feature derived from the edge map $\mathbf{E}_o$, $\mathbf{c}_i$ is the 2D coordinate of pixel i on image. $\omega_1, \omega_2, \theta_\alpha, \theta_\beta, \theta_\gamma$ are parameters.

To transform the pairwise similarity derived from the edge map to the feature representation $\mathbf{f}$ for efficient computing, we borrow the idea from the Normalized Cut (NCut) for segmentation [SM00, Kok16a], where we can first generate an affinity matrix between pixels using intervening contour [SM00], and perform normalized cut. We select the top 6 dimensions from the resultant Eigen vectors as our feature $\mathbf{f}$ for each pixel. A transformation from edge to the planar affinity using the Eigen vectors is shown in Fig. 4.3. As can be seen from the affinity map, the NCut features are

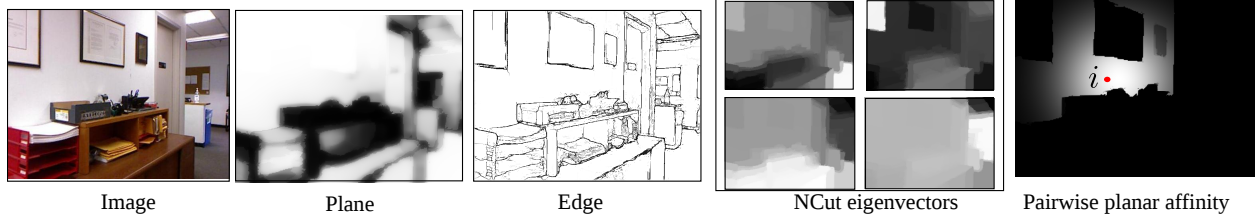


Figure 4.3: Pairwise surface affinity from the plane and edge predictions with computed Ncut features. We highlight the computed affinity w.r.t. pixel i (red dot).

effective to determine whether two pixels lie in the same planar surface where the regularization can be performed.

4.4 Optimization

Given the formulation in Sec. 4.3, we first discuss the fast inference implementation for DCRF, and then present the algorithm of joint training with CNNs through back-propagation.

4.4.1 Inference

To optimize the objective function defined in Eqn.(4.1), we use mean-field approximation for fast inference as used in the optimization of DCRF [KK12a]. In addition, we chose to use coordinate descent to sequentially optimize normals and depth. When optimizing normals, for simplicity and efficiency, we do not consider the term of $\mu_d()$ in Eqn.(4.3), yielding the updating for pixel i at iteration t as,

$$\mathbf{n}_i^{(t)} \leftarrow \frac{1}{2} w_i^n \mathbf{n}_i^o + \frac{\lambda}{2} \sum_{j, j \neq i} w_j^n \mathbf{n}_j^{(t-1)} \mathbf{A}_{i,j}, \quad \mathbf{n}_i^{(t)} \leftarrow \mathbf{n}_i^{(t)} / \|\mathbf{n}_i^{(t)}\|_2, \quad (4.6)$$

which is equivalent to first performing a dense bilateral filtering [BP15] with our pairwise planar affinity term $\mathbf{A}_{i,j}$ for the predicted normal map, and then applying L2 normalization.

Given the optimized normal information, we further optimize depth values. Similar to normals, after performing mean-field approximation, the inferred updating equation for depth at iteration t is,

$$d_i^{(t)} \leftarrow \frac{1}{\nu_i} \left(w_i^d d_i^o + \lambda (\mathbf{n}_i \cdot \mathbf{p}_i) \sum_{j, j \neq i} \mathbf{A}_{i,j} w_j^d d_j^{(t-1)} (\mathbf{n}_j \cdot \mathbf{p}_j) \right) \quad (4.7)$$

where $\nu_i = w_i^d + \lambda(\mathbf{n}_i \cdot \mathbf{p}_i) \left(\mathbf{p}_i \cdot \sum_{j,j \neq i} \mathbf{A}_{i,j} w_j^d \mathbf{n}_j \right)$, Since the graph is densely connected, previous work [KK12b] indicates that only a few (<10) iterations are need to achieve reasonable performance. In practice we found that 5 iterations for normal inference and 2 iterations for depth inference yielded reasonable results.

4.4.2 Joint training of CNN and DCRF

We further implement the DCRF inference as a trainable layer as in [ZJR15] by considering the inference as feedforward process, to enable joint training together with the normal and depth neural networks. This makes the planar surface information able to be back-propagated to the neural networks and further refine their output. We describe the gradients back-propagated to the two networks respectively.

Back-propagation to the normal network. Suppose the gradient of normal passed from the upper layer after DCRF for pixel i is $\nabla_f(\mathbf{n}_i)$, which is a 3×1 vector. We now back-propagate it first through the L2 normalization using the equation of $\nabla_{L2}(\mathbf{n}_i) = (\mathbf{I}/\|\mathbf{n}_i\| - \mathbf{n}_i \mathbf{n}_i^T / \|\mathbf{n}_i\|^3) \nabla_f(\mathbf{n}_i)$, and then back-propagate through the mean-field approximation in Eqn. (4.6) as,

$$\frac{\partial L(\mathbf{N})}{\partial \mathbf{n}_i} = \frac{\nabla_{L2}(\mathbf{n}_i)}{2} + \frac{\lambda}{2} \sum_{j,j \neq i} \mathbf{A}_{j,i} \nabla_{L2}(\mathbf{n}_j), \quad (4.8)$$

where $L(\mathbf{N})$ is the loss from normal predictions after DCRF, $\mathbf{I}$ is the identity matrix.

Back-propagation to the depth network. Similarly for depth, suppose the gradient from the upper layer is $\nabla_f(d_i)$, the depth gradient for back-propagation through Eqn. 4.7 can be inferred as,

$$\frac{\partial L(\mathbf{D})}{\partial d_i} = \frac{1}{\nu_i} \nabla_f(d_i) + \lambda(\mathbf{n}_i \cdot \mathbf{p}_i) \sum_{j,j \neq i} \frac{1}{\nu_j} \mathbf{A}_{j,i} (\mathbf{n}_j \cdot \mathbf{p}_j) \nabla_f(d_j) \quad (4.9)$$

where $L(\mathbf{D})$ is the loss from depth predictions after DCRF.

Note that during back propagation for both surface normals and depths we drop the confidences w since using it during training will make the process very complicated and inefficient. We adopt the same surface normal and depth loss function as in [EF15] during joint training. Also, it is possible to also back propagate the gradients of the depth values to the normal network via the

surface normal and depth compatibility in Eqn. (4.4). However, this involves the depth values from all the pixels within the same plane, which may be intractable and cause difficulty during joint learning. We therefore chose not to back propagate through the compatibility in our current implementation and leave it to future work.

4.5 Implementation details for DCRF

To predict the input surface normals and depths, we build on the publicly-available implementation from Eigen and Fergus [EF15], which is at or near state of the art for both tasks. We compute prediction confidences for the surface normals and depths using Monte Carlo dropout [SHK14]. Specifically, we forward propagate through the network 10 times with dropout during testing, and compute the prediction variance v_i at each pixel. The predictions with larger variance v_i are considered less stable, so we set the confidence as $w_i = \exp(-v_i/\sigma.^2)$. We empirically set $\sigma_n = 0.1$ for normals prediction and $\sigma_d = 0.15$ for depth prediction to produce reasonable confidence values.

Specifically, for prediction the plane map $\mathbf{P}_o$, we adopt a semantic segmentation network structure similar to the Deeplab [CPK15] network but with multi-scale output as the FCN [LSD15b]. The training is formulated as a pixel-wise two-class classification problem (planar vs. non-planar). The output of the network is hereby a plane probability map $\mathbf{P}_o$ where p_i at pixel i indicates the probability of pixel i belonging to a planar surface. The edge map $\mathbf{E}_o$ indicates the plane boundaries. During training, the ground-truth edges are extracted from the corresponding ground-truth depth and normal maps, and refined by semantic annotations when available (see Fig.4.4 for an example). We then adopt the recent Holistic-nested Edge Detector (HED) network [XT15] for training. In addition, we augment the network by adding predicted depth and normal maps as another 4-channel input to improve recall, which is very important for our regularization since missing edges could mistakenly merge two planes and propagate errors during the message passing.

For the surface bilateral filter in Eqn. (4.5), we set the parameters $\theta_\alpha = 0.1, \theta_\beta = 50, \theta_\gamma = 3, \omega_1 = 1, \omega_2 = 0.3$, and set the $\lambda = 2$ in Eqn.(4.1) through a grid search over a validation set from [GGA14]. The four types of inputs to the DCRF are aligned and resized to 294x218 by matching the network output of [EF15]. During the joint training of DCRF and CNNs, we fix the

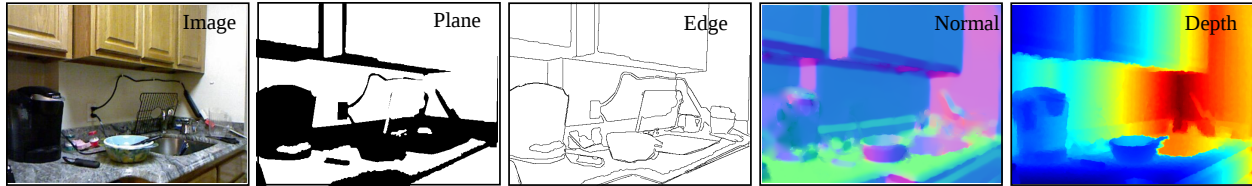


Figure 4.4: Four types of ground-truth from the NYU dataset that are used in our algorithm.

parameters and fine-tune the network based on the weights pre-trained from [EF15], with the 795 training images, and use the same loss functions and learning rates as in their depth and normal networks respectively.

Due to limited space, the detailed edge and plane network structures, the learning and inference times and visualization of confidence values are presented in the supplementary materials.

4.6 Experiments

We perform all our experiments on the NYU v2 dataset [SHK12]. It contains 1449 images with size of 640×480 , which is split to 795 training images and 654 testing images. Each image has an aligned ground-truth depth map and a manually annotated semantic category map. In addition, we use the ground-truth surface normals generated by [LZP14] from depth maps. We further use the official NYU toolbox¹ to extract planar surfaces from the ground-truth depth and refine them with the semantic annotations, from which a binary ground-truth plane map and an edge map are obtained. The details of generating plane and edge ground-truth are elaborated in supplementary materials. Fig. 4.4 shows the produced four types of ground-truth maps for our learning and evaluation.

We implemented all our algorithms based on Caffe [JSD14], including DCRF inference and learning, which are adapted from the implementation in [ABD10, ZJR15].

Evaluation setup. In the evaluation, we first compare the normals and depths generated by different baselines and components over the ground truth planar regions, since these are the regions where we are trying to improve, which are most important for 3D editing applications. We evaluated

¹http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

over the valid 561x427 area following the convention in [LSL15a, LZP14]. We also perform evaluation over the ground truth edge area showing that our results preserve better geometry details. Finally, we show the improvement achieved by our algorithm over the entire image region.

We compare our results against the recent work Eigen et.al [EF15] since it is or is near state-of-the-art for both depth and normal. In practice, we use their published results and models for comparison. In addition, we implemented a baseline method for hard planar regularization, in which the planar surfaces are explicitly extracted from the network predictions. The normal and depth values within each plane are then used to fit the plane parameters, from which the regularized normal and depth values are obtained. We refer to this baseline as "Post-Proc.". For normal prediction, we implemented another baseline in which a basic Bilateral filter based on the RGB image is used to smooth the normal map.

In terms of the evaluation criteria, we first adopt the pixel-wise evaluation criteria commonly used by previous works [WSL15a, EF15]. However, as mentioned in [HMK15], such metrics mainly evaluate pixel-wise depth and normal offsets, but do not well reflect the quality of reconstructed structures over edges and planar surfaces. Thus, we further propose plane-wise metrics that evaluate the consistency of the predictions inside a ground truth planar region. In the following, we first present evaluations for normal prediction, and then report the results of depth estimation.

Surface normal criteria. For pixel-wise evaluation, we use the same metrics used in [EF15].

For plane-wise evaluation, given a set of ground truth planar regions $\{\mathbf{P}_j^*\}_{j=1}^{N_P}$, we propose two metrics to evaluate the consistency of normal prediction within the planar regions,

1. Degree variation (var.): It measures the overall planarity inside a plane, and defined as, $\frac{1}{N_P} \sum_j \frac{1}{|\mathbf{P}_j^*|} \sum_{i \in \mathbf{P}_j^*} \delta(\mathbf{n}_i, \bar{\mathbf{n}}_j)$, where $\delta(\mathbf{n}_i, \mathbf{n}_j) = \text{acos}(\mathbf{n}_i \cdot \mathbf{n}_j)$ which is the degree difference between two normals, $\bar{\mathbf{n}}_j$ is the normal mean of the prediction inside $\mathbf{P}_j^*$.
2. First-order degree gradient (grad.): It measures the smoothness of the normal transition inside a planar region. Formally, it is defined as, $\frac{1}{N_P} \sum_j \frac{1}{|\mathbf{P}_j^*|} \sum_{i \in \mathbf{P}_j^*} (\delta(\mathbf{n}_i, \mathbf{n}_{hi}) + \delta(\mathbf{n}_i, \mathbf{n}_{vi}))$, where $\mathbf{n}_{hi}, \mathbf{n}_{vi}$ are normals of right and bottom neighbor pixels of i .

Evaluation on surface normal estimation. In upper part of Tab. 4.1, we show the comparison results. The first line, i.e. Eigen-VGG, is the result from [EF15] with VGG net, which serves as our baseline. The simple RGB-Bilateral filtering can only slightly improve the network output since it does not contain any planar surface information during the smoothing. The hard regularization over planar regions ("Post-Proc.") can improve the plane-wise consistency since hard constraints are enforced in each plane, but it also brings strong artifacts and suffers significant decrease in pixel-wise accuracy metrics. Our "DCRF" can bring improvement on both pixel-wise and plane-wise metrics, while integrating network prediction confidence further makes the DCRF inference achieve much better results. Specifically, using "DCRF-conf", the plane-wise error metric var. drops from 9.15 produced by the network to 6.8. It demonstrates that our non-local planar surface regularization does help the predictions especially for the consistency inside planar regions.

We also show the benefits from the joint training of DCRF and CNN. "Eigen-VGG (JT)" denotes the output of the CNN after joint training, which shows better results than the original network. It indicates that regularization using DCRF for training also improves the network. By using the joint trained CNN and DCRF ("DCRF (JT)"), we observe additional improvement over that from "DCRF". Finally, by combining the confidence from joint trained CNN, our final outputs ("DCRF-conf (JT)") achieve the best results over all the compared methods. In addition, we also use ground-truth plane and edge map to regularize the normal output("Oracle") to get an upper bound when the planar surface information is perfect. We can see our final results are in fact quite close to "Oracle", demonstrating the high quality of our plane and edge prediction.

In the bottom part of Tab. 4.1, we show the evaluation over edge areas (rows marked by "Edge") as well as on the entire images (marked by "Image"). The edge areas are obtained by dilating the ground truth edges with 10 pixels. Compared with the baseline, although our results slightly drop in "mean" and 30° , they are much better in "median" and 11.25° . It shows by preserving edge information, our geometry have more accurate predictions around boundaries. When evaluated over the entire images, our results outperforms the baseline in all the metrics, showing that our algorithm not only largely improves the prediction in planar regions, but also keeps the good predictions within non-planar regions.

	Pixel-wise (Over planar region)					Plane-wise	
	Lower the better		Higher the better			Lower the better	
Evaluation over the planar regions							
Method	mean	median	11.25°	22.5°	30°	var.	grad.
Eigen-VGG [EF15]	14.5425	8.9735	59.00	80.85	87.38	9.1534	1.1112
RGB-Bilateral	14.4665	8.9439	59.12	80.86	87.41	8.6454	1.1735
Post-Proc.	14.8154	8.6971	59.85	80.52	86.67	7.2753	0.9882
Eigen-VGG (JT)	14.4978	8.9371	59.12	80.90	87.43	8.9601	1.0795
DCRF	14.1934	8.8697	59.27	81.08	87.77	6.9688	0.7441
DCRF (JT)	14.2055	8.8696	59.34	81.13	87.78	6.8866,	0.7302
DCRF-conf	13.9732	8.5320	60.89	81.87	88.09	6.8212	0.7407
DCRF-conf (JT)	13.9763	8.2535	62.20	82.35	88.08	6.3939	0.6858
Oracle	13.5804	8.1671	62.83	83.16	88.85	4.9199	0.5923
Eigen-VGG [EF15]	23.4141	18.3288	30.90	58.91	71.43	Edge	
DCRF-conf (JT)	23.4694	17.6804	33.63	59.53	71.03		
Eigen-VGG [EF15]	20.9322	13.2214	44.43	67.25	75.83	Image	
DCRF-conf (JT)	20.6093	12.1704	47.29	68.92	76.64		

Table 4.1: Normal accuracy comparison over the NYU v2 dataset. We compare our final results (DCRF-conf (JT)) against various baselines over ground truth planar regions at upper part, where JT means joint training CNN and DCRF as presented in Sec. 4.4.2. We list additional comparison over the edge and full image region at lower part.

Depth criteria. When evaluating depths, similarly, we also firstly adopt the traditional pixel-wise depth metrics that are defined in [EPF14, WSL15a]. We refer readers to the original papers for detailed definition due to limited space. We then also propose plane-wise metrics. Specifically, we generate the normals from the predicted depths using the NYU toolbox [SHK12], and evaluate the degree variation (var.) of the generated normals within each plane.

Evaluation on depth prediction. Similarly, we first report the results on planar regions in the upper part of Tab. 4.2, and then present the evaluation on edge areas and over the entire image. We can observe similar trends of different methods as in normal evaluation, demonstrating the effectiveness of the proposed approach in both tasks.

	Pixel-wise						Plane-wise		
	Lower the better (LTB)					Higher the better			LTB
Evaluation over the planar regions									
Method	Rel	Rel(sqr)	$\log_{10}$	$RMSE_{lin}$	$RMSE_{log}$	1.25	1.25^2	1.25^3	var.
Eigen-VGG [EF15]	0.1441	0.0892	0.0635	0.5083	0.1968	78.7055	96.3516	99.3291	16.4460
Post-Proc.	0.1470	0.0937	0.0644	0.5200	0.2003	78.2290	96.1145	99.2258	11.1489
Eigen-VGG(JT)	0.1427	0.0881	0.0612	0.4900	0.1930	80.1163	96.4421	99.3029	17.5251
DCRF	0.1438	0.0893	0.0634	0.5100	0.1965	78.7311	96.3739	99.3321	12.0424
DCRF(JT)	0.1424	0.0874	0.0610	0.4873	0.1920	80.1800	96.5481	99.3326	10.5836
DCRF-conf	0.1437	0.0881	0.0631	0.5027	0.1957	78.9070	96.4336	99.3395	12.0420
DCRF-conf(JT)	0.1423	0.0874	0.0610	0.4874	0.1920	80.2453	96.5612	99.3229	10.5746
Oracle	0.1431	0.0879	0.0629	0.5043	0.1950	78.9777	96.4297	99.3605	8.0522
Eigen-VGG [EF15]	0.1645	0.1369	0.0735	0.7268	0.2275	72.9491	94.2890	98.6539	Edge
DCRF-conf(JT)	0.1624	0.1328	0.0707	0.6965	0.2214	74.7198	94.6927	98.7048	
Eigen-VGG [EF15]	0.1583	0.1213	0.0671	0.6388	0.2145	77.0536	95.0456	98.8140	Image
DCRF-conf(JT)	0.1555	0.1179	0.0672	0.6430	0.2139	76.8466	95.0946	98.8668	

Table 4.2: Depth accuracy comparison over the NYU v2 dataset.

Qualitative results. We also visually show an example to illustrate the improvements brought by our method. In Fig. 4.5, we visualize the predictions in 3D space in which the reconstructed structure can be better observed. As can be seen, the results from network output [EF15] have lots of distortions in planar surfaces, and the transition is blurred accross plane boundaries, yielding non-satisfactory quality. Our results largely allievate such problems by incorporating plane and edge regularization, yielding visually much more satisfied results. Due to space limitation, we include more examples in the supplementary materials.

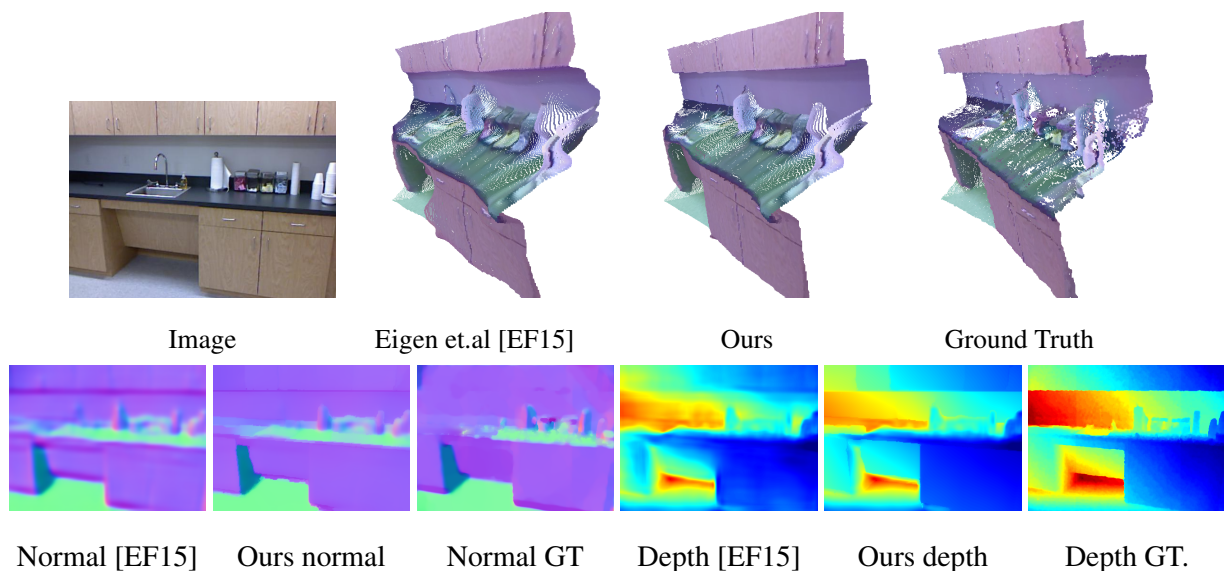


Figure 4.5: Visual comparison between network output from Eigen et.al [EF15] and our results in 3D view. We project the RGB and normal color to the 3D points (Best view in color).

4.7 Conclusion

In this paper, we introduce SURGE, which is a system that induces surface regularization to depth and normal estimation from a single image. Specifically, we formulate the problem as DCRF which embeds surface affinity and depth normal compatibility into the regularization. Last but not the least, our DCRF is enabled to be jointly trained with CNN. From our experiments, we achieve promising results and show such regularization largely improves the quality of estimated depth and surface normal over planar regions, which is important for 3D editing applications.

4.8 Appendix for SURGE

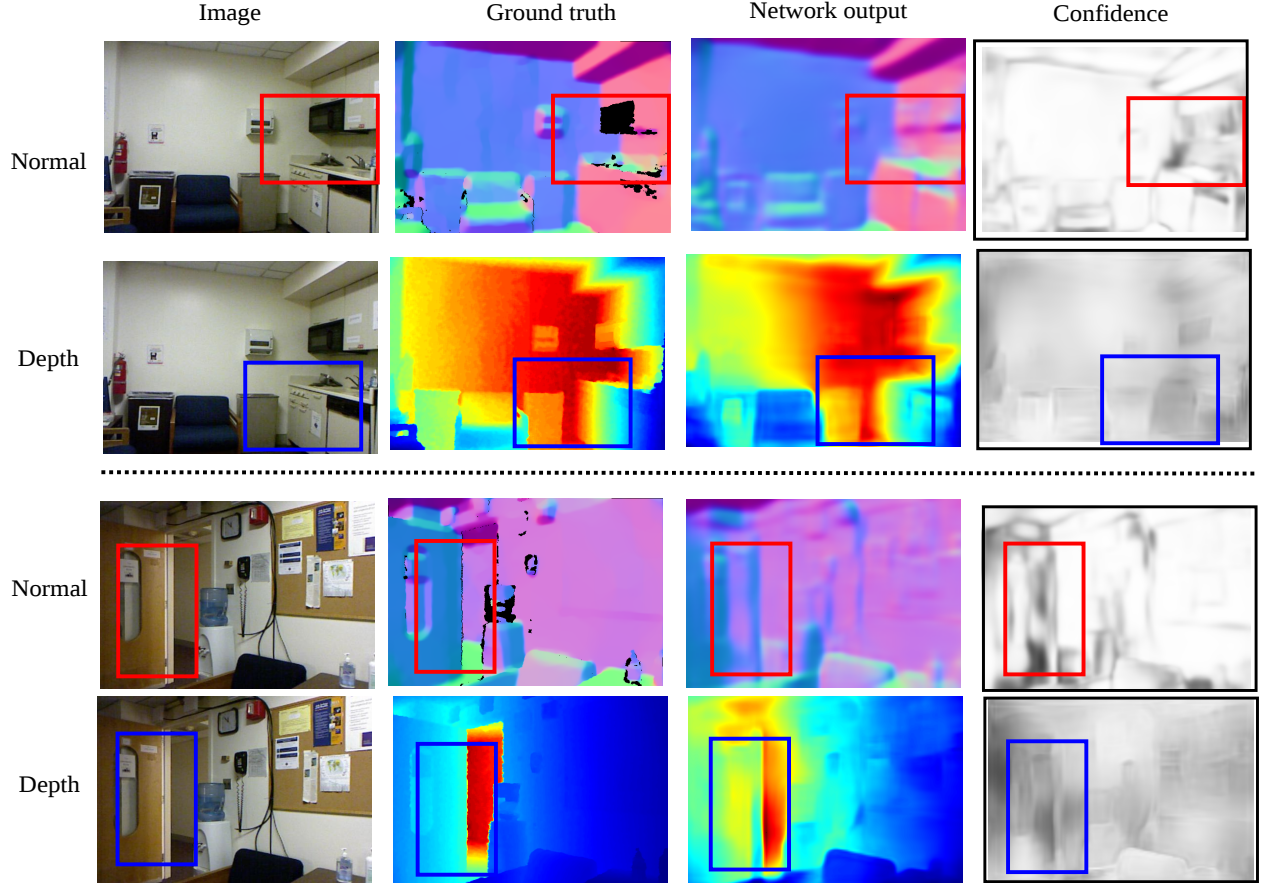


Figure 4.6: Example to visualize of the prediction variation which can be used to represent prediction confidence by using the Monte Carlo dropout (last column). We highlighted place where confidence indicate errors for depth and normal.

4.8.1 Visualization of network prediction confidence

In Fig. 4.6, we illustrate two examples with the confidence used in our prediction for both depth and normal. We highlight the places where prediction errors are indicated by the confidence. We found that the normal confidence is visually more accurate, while the depth confidence is relatively more unified with less variation. This is consistent with our experiments that incorporating normal confidence provides more performance gains, as shown in Tab.1 and Tab.2 respectively in the experiments.

4.8.2 Details of extracting ground-truth planar surfaces

To extract the plane regions, we first obtain the planar candidates by processing the ground-truth depth using NYU toolbox planar extraction [LZP14], which is defined as minimum number (2500) of contiguous pixels that can be fit into a single 3D plane after projecting their depth values into 3D world. We also obtain the semantic regions from the semantic annotations. For each semantic instance region $\mathbf{R}_i$, we intersect it with the planar candidate regions, and find a set with all the intersected plane regions inside, which we denote as $\mathcal{S}_{\mathbf{R}_i} = \{\mathbf{P}_j | \mathbf{P}_j \subseteq \mathbf{R}_i\}$. Then if any region in $\mathcal{S}_{\mathbf{R}_i}$ covers 90% of the region of $\mathbf{R}_i$, we let $\mathcal{S}_{\mathbf{R}_i} = \{\mathbf{R}_i\}$. We collect all such regions from the image and generate planar instance set, i.e. $\mathcal{P} = \cup_i \mathcal{S}_{\mathbf{R}_i}$.

4.8.3 Network structures for plane and edge prediction.

We present the details of edge and plane neural network architectures. Fig. 4.7 illustrates the edge network structure. It has two streams. The bottom stream takes the RGB image as input, and has the same architecture as in HED [XT15], while the top stream is a very shallow one, which takes the predicted depth and normal maps as input to augment the prediction from images. In particular, it has a 64-channel convolutional layer with kernel size 9, followed by a fully-connected layer to generate the predictions. The predictions are then fuse into all the five side output in the HED stream.

Fig. 4.8 illustrates the structure of plane network. It follows the multi-scale structure with side output from "conv3", "conv4" and "conv5" as in the fully convolutional network [LSD15b], but with a light-weighted network structure same with the Deeplab-LargeFOV [CPK15].

4.8.4 Learning, inference time

The inference includes performing 4-stream network forward propagation: 1.4s for depth and normal (320×240 input), 1.5s for edge (640×480) and plane (320×240) with Titian X GPU, 10s for normalized cut by our Matlab implementation (320×240 input), which could be substantially faster (0.2s) using GPU implementation from [CSS09]. The final DCRF takes 1s for inference with

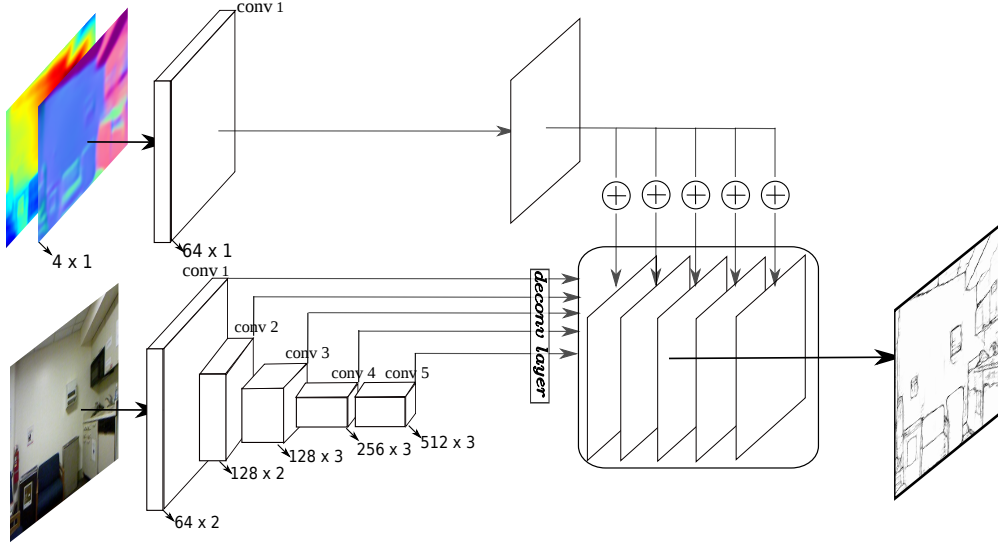


Figure 4.7: Architecture of edge network with two input. At the corner of each layer we show the number channel $\times$ the number of sub-layers.

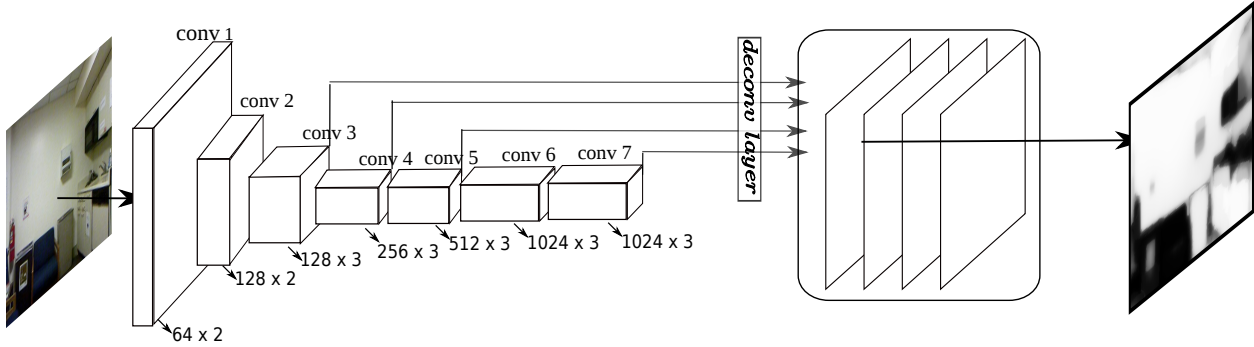


Figure 4.8: Architecture of plane network that takes RGB image and output a binary plane map.

a single CPU (3.2GHz). In total, it takes around 14s for each image with (640×320) in our current implementation. For learning our system, it takes around 10 hours each to train an edge and a plane network, and takes around 2 days for fine-tuning over depth and normal networks due to the CPU implementation of DCRF layer.

4.8.5 Additional qualitative comparison from NYU v2 data.

In Fig. 4.9, Fig. 4.10, Fig. 4.11, we show more examples to visually demonstrate the improvement over the planar surface regions. For each example, we show overhead 3D view at the first row, and

corresponding normal and depth images at the second row. The results from left to right are: Eigen et.al [EF15], ours and ground truth.

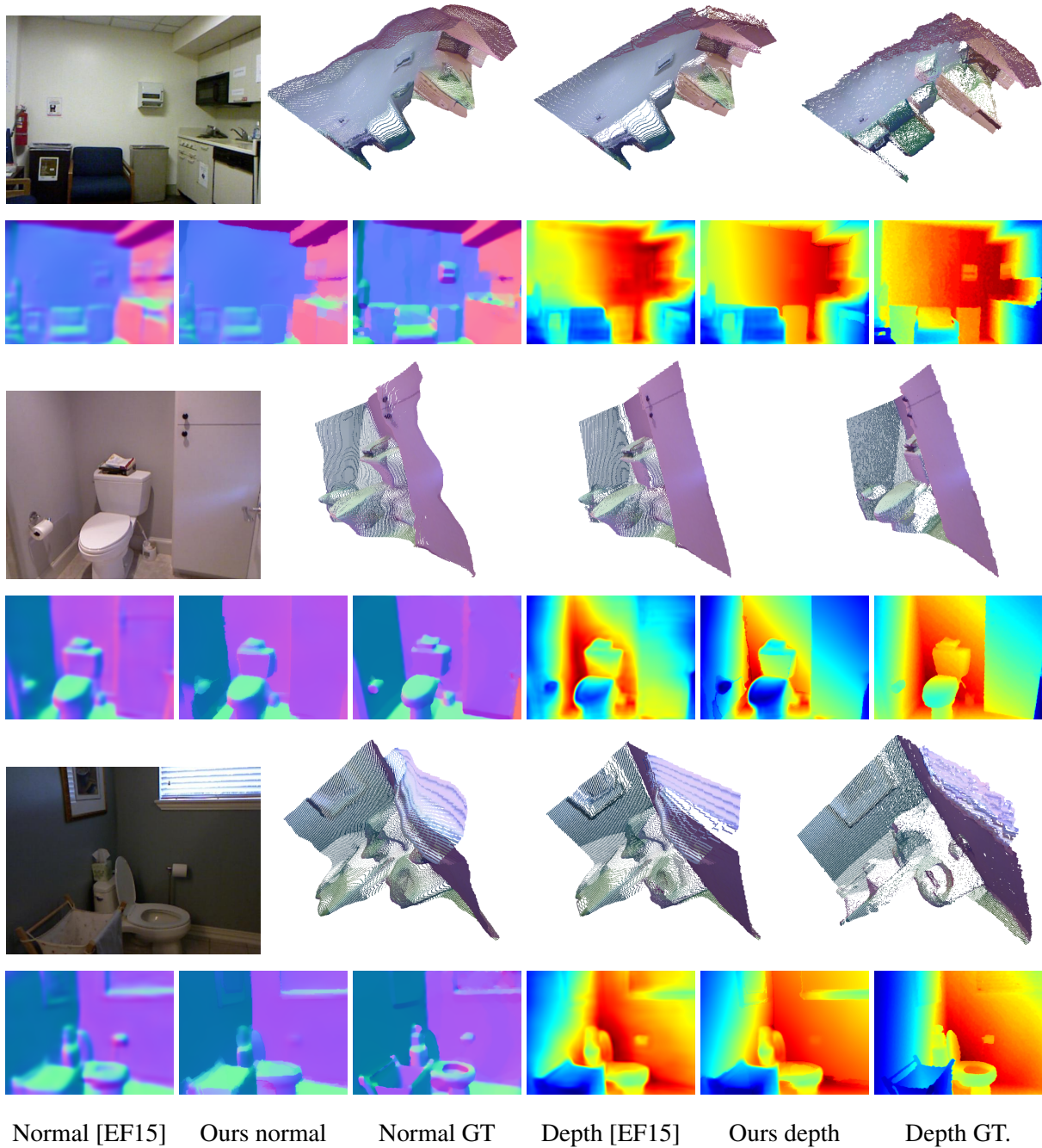


Figure 4.9: Additional visual comparison between network output from Eigen et.al [EF15] and our results (Best view in color) from NYU v2 dataset.

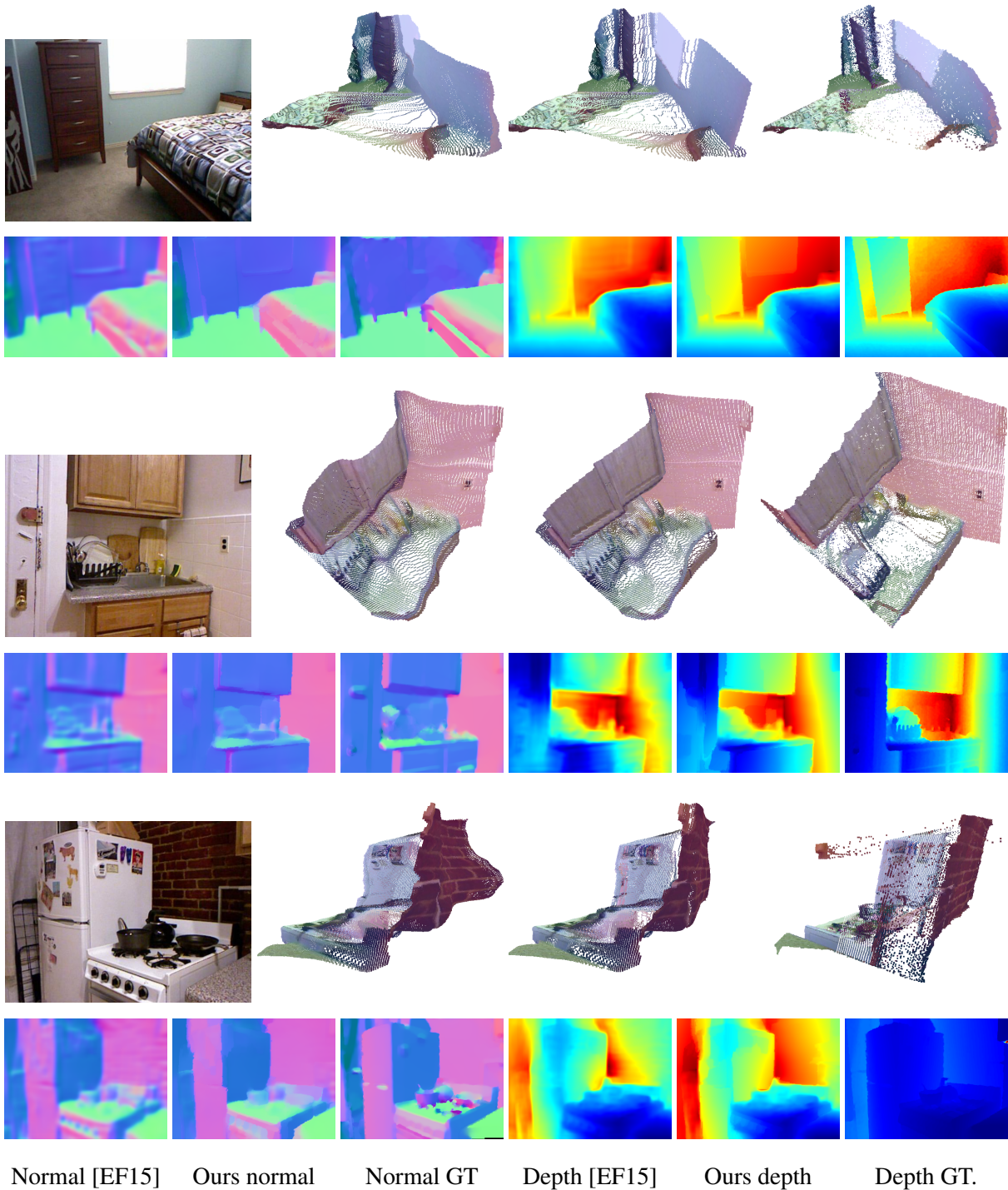


Figure 4.10: Additional visual comparison between network output from Eigen et.al [EF15] and our results (Best view in color) from NYU v2 dataset.

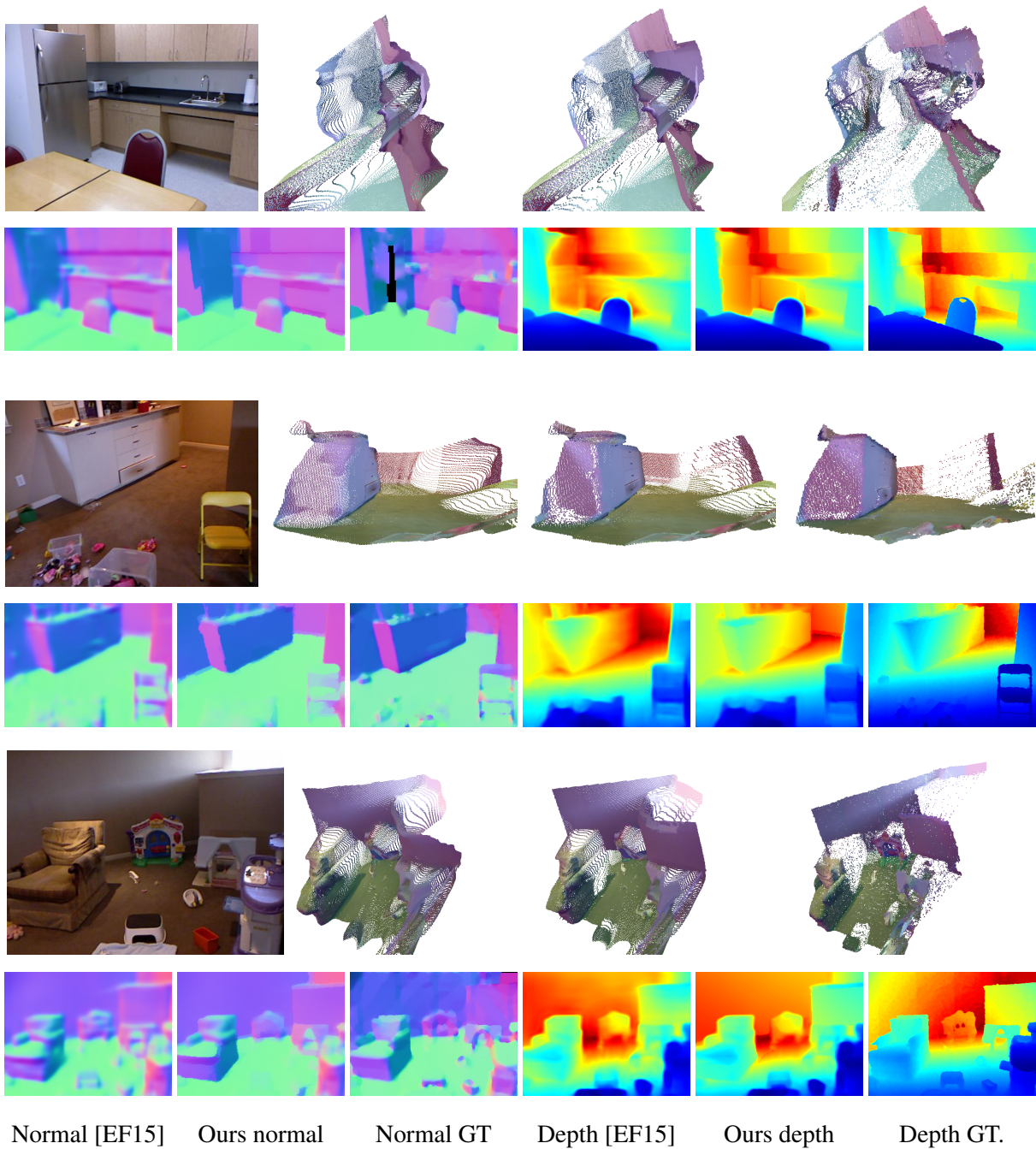


Figure 4.11: Additional visual comparison between network output from Eigen et.al [EF15] and our results (Best view in color) from NYU v2 dataset.

CHAPTER 5

Joint part and object segmentation with deep learned potentials

Decomposing an object into semantic parts enables a more detailed understanding of the object, which can provide additional information to benefit many computer vision tasks such as pose estimation [YR11, DCY14], detection [AL12, CML14], segmentation [EW12], and fine-grained recognition [ZDG14]. Thus, it has become an attractive research topic to leverage semantic part representation through part detection [BM09b, FGM10, CML14], and human joint estimation [TS14].

In the literature of semantic segmentation, while object-level segmentation over multiple object categories has been extensively studied along with the growing popularity of standard evaluation benchmarks such as PASCAL VOC [EVW10], object parsing (i.e., segmenting objects into semantic parts) is addressed mostly for a few specific categories provided with accurate localization such as human [ZCL11, EW12, DCY14] and cars [EW12].

With the increasing availability of semantic part annotations [CML14], more recent works have attempted to handle more difficult classes like animals with homogeneous appearance [WY15], and to perform both object and part segmentation [HAG15], as illustrated in Fig. 5.1. However, in [HAG15], object and part segmentation are performed sequentially, in which the object mask is first segmented, and then the part labels are assigned to the pixels within the mask. As a result, the errors from the predicted semantic object masks may be propagated to the parts.

In fact, object and part segmentation are complementary and mutually beneficial to each other. Semantic object segmentation requires a larger receptive field in order to correctly recognize the object, while part segmentation focuses on local details to obtain more accurate segmentation boundaries and accommodate large pose and viewpoint variations. If these two tasks are tackled simultaneously, by integrating the object-level guidance with part-level detailed segmentation, we can address two of the most challenging problems at the same time, i.e., discovering the subtle

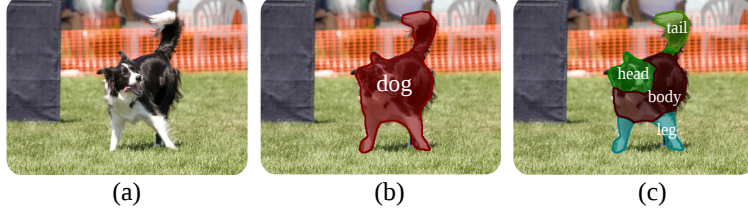


Figure 5.1: We handle the prediction of semantic object and part segmentation in a wild scene scenario. (a) Original image. (b)&(c) are the object and part segmentation respectively, generated from our algorithm.

appearance differences between different parts within a single object, and avoiding the ambiguity across similar object categories. Motivated by this observation, we propose a joint solution to object and part segmentation, in which the consistency of the object and parts are enforced through joint training and inference.

Fig. 5.2 shows the overall framework of our approach. When performing part segmentation over multiple object classes, the appearance of some parts may be very similar, *e.g.*, horse legs and cow legs. Therefore in order to reduce the ambiguity and complexity during training, instead of treating each semantic part type independently [HAG15], we allow some part labels to be shared by related object classes, and group the labelled parts of different classes into a semantic compositional part (SCP) representation based on their appearance and shape similarity (*e.g.*, horse legs and cow legs belonging to the same type of leg). Details of part sharing and SCP labels are illustrated in Fig. 5.3. Since we are performing object and part segmentation jointly, the ambiguity of shared part labels can be solved by the object label.

Given an image with certain objects, we train a two-stream fully convolutional network (FCN) with the first channel predicting the SCP potentials while the second predicting the object potentials. We then concatenate the two potentials as the input to an additional convolutional layer to refine the object potentials through joint training. At the same time, a compact set of SCP region proposals are generated from the SCP potentials. Using these region proposals as nodes, we construct a fully connected conditional random field (FCRF) to further incorporate the object and SCP potentials, yielding the jointly predicted object and part segmentations. In our FCRF, the consistency between

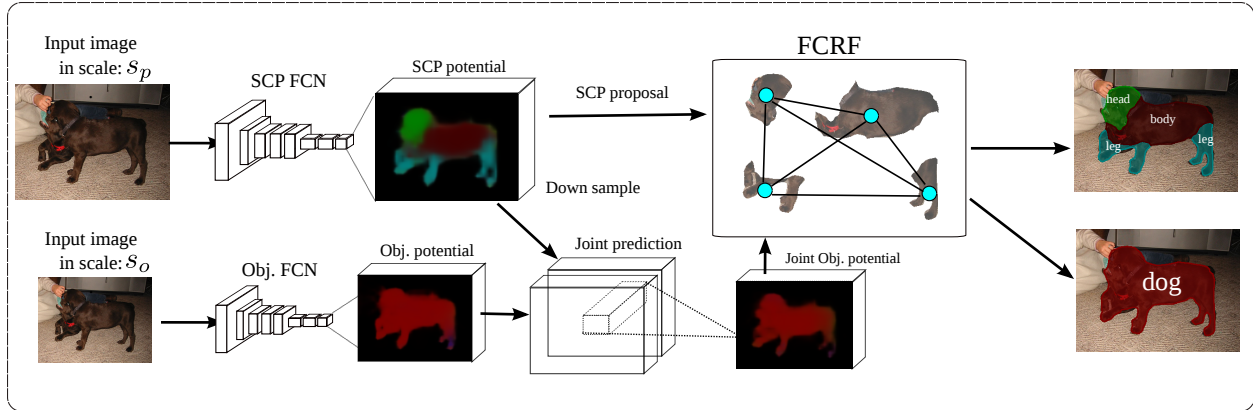


Figure 5.2: Our framework for joint part and object segmentation. Given an image, a two-stream FCN is performed to predict both the semantic compositional parts (SCP) and object potentials with different input image scales. The two potentials are then concatenated and fed to a new convolutional layer to predict joint object potentials. Finally, from SCP potentials, SCP segments are proposed as nodes for the fully-connected CRF to jointly infer the part and object labels.

the object and parts are enforced with long-range constraints.

We did extensive experiments over three derived datasets based on PASCAL VOC segmentation benchmark [EVWa, CML14]. Experimental results demonstrate that our joint approach improves both object and part segmentation, and significantly outperforms the state-of-the-art on both tasks.

5.1 Related work

In the literature of detection, the usefulness of mining semantic part representation in helping object recognition has been long studied. Felzenszwalb et.al [FGM10, GFM11] proposed the deformable part-based model (DPM) which is an implicit way of discovering hidden parts. Later models use more accurate part-representation through explicit part supervision [SS11, ZCT10]. Poselets [BM09b] are proposed to model the local human parts through 2D projections from 3D data, which can be used as robust representation for both detection [GAL12] and segmentation [MYP11, AHG12]. In addition to human body parts, Azizpour et. al [AL12] explicitly induce the bounding-box annotations of animal parts, yielding stronger detection results on both object and parts. Chen

et. al [CML14] extend such ideas by providing richer labeling of part segments, which can better capture the appearance features for learning.

In the literature of segmentation, semantic parsing has also been actively investigated. However, due to its increased challenge in getting detailed boundaries, most previous work focused on parsing objects given both the semantic category and a cropped bounding box with no occlusion, such as human parsing [BF11, ZCL11, YKO12, EW12, DCY14], car parsing [TFL08, EW12, LLY14] or animal parsing [WY15]. Such methods are limited in their applications, as objects in real-world images are often occluded with large deformation and appearance variations, which is difficult to be handled by those shape-based [BF11] or appearance-based [DCY14] models with hand-crafted features or bottom-up segments.

Recently, deep convolutional neural networks [KSH12] (CNN) have achieved great success in many applications such as object detection [GDD14, HAG14, ZUS15] and end-to-end segmentation [WSL15a, HAG15, LSD15b, CPK15], with advanced network structures such as the VGG-Net [SZ14]. Some studies tried to understand the implicitly learned filters [ZF14] or comparable structures with the DPM [WEF14, GID15] in the network, and discovered some meaningful part representations in deeper layers. However, such representations are still not semantic parts, and using explicit supervision from semantic part labels for segmentation has not been investigated. In our approach, we propose to explicitly model the semantic parts along with the whole object by taking advantage of the recent advance of fully convolutional network (FCN) [LSD15b]. It has been very successful in predicting structured output such as semantic object segmentation. Specifically, FCN converts the fully-connected layers in the original CNN to 1×1 convolution layers, thus can efficiently perform sliding-window-based classification at each pixel with a certain receptive field. However, it starts from local convolutional kernels with limited receptive fields and may not be able to capture all the long-range context, yielding local confusions. In our case, we solve such problem through modelling over object scale context, which is also suggested in prior arts [BM09b, DHH09, TB10, DGE14].

Perhaps the closest work in our scenario is the hypercolumn approach [HAG15]. However, they perform object and part segmentation sequentially, and train many part classifiers separately for each class, which may suffer from increased training cost and has less scalability when the

number of object class is large. In contrast, we use semantic compositional part (SCP) to allow part sharing and reduce training complexity. Moreover, our model leverages the advantage of both object segmentation and semantic part segmentation, yielding strong results in very challenging scenarios. To the best of our knowledge, this is the first work that provides a joint solution to tackle the segmentation of semantic parts and objects, which allows the part and object potentials interact and benefit each other.

5.2 Joint part and object segmentation

Our framework includes four major parts, i.e. shared semantic compositional parts (SCP) generation, part and object potentials, proposal of SCP regions and fully connected conditional random field (FCRF). In the following sessions, we will describe these techniques in details.

5.2.1 Semantic compositional parts

When performing semantic object segmentation and parsing over multiple object classes, some parts from different object classes yet with similar semantic meanings may also have very similar shapes and appearances (*e.g.*, horse legs and cow legs). In such scenarios, allowing the parts to be shared among similar object classes [ZCT10, OE11] could alleviate the difficulties of distinguishing similar parts from different objects, and at the same time reduce the increasing complexity of training and inference as the number of object categories grows. Therefore, before we formally train our framework for these two tasks, we group those similar parts to form the semantic compositional parts (SCP) that are shared among related object classes.

In particular, given a semantic part, it has an object label l_o (*e.g.*, horse) and a particular semantic meaning l_s (*e.g.*, leg). The joint label of this part we would like to infer is denoted by l_{op} (*e.g.*, horse-leg). We group the original part labels l_{op} to a shared compositional part representation l_{scp} if they have the same semantic meanings l_s and highly similar appearances and shapes. The SCP l_{scp} are then used to compose different objects l_o as illustrated in Fig. 5.3. For example, in the case of horse and cow, the representation of different labels are, $l_o \in \{\text{horse, cow}\}$, $l_{op} \in \{\text{horse-head,}$

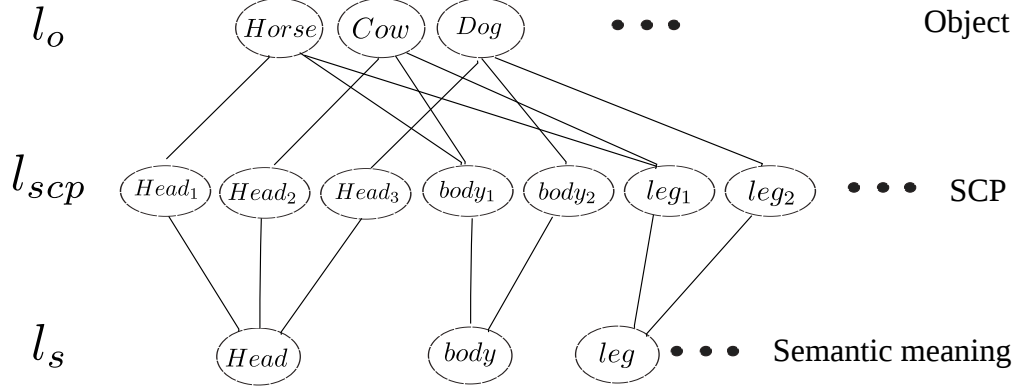


Figure 5.3: Illustration of our semantic compositional part (SCP) grammar in Sec. 5.2.1. Each SCP is associated with one semantic meaning, and all the objects are composed of several SCPs.

horse-body, horse-leg, horse-tail, cow-head, cow-leg, cow-body, cow-tail} and $l_s \in \{\text{head, body, leg, tail}\}$. If we allow the two objects to share the same type of body, leg and tail, we get the SCP label $l_{scp} \in \{\text{head(h), head(c), body}_1, \text{leg}_1, \text{tail}_1\}$, as in Fig. 5.3, which is a much smaller prediction space than l_{op} . The information of l_{op} is kept in the connections between l_o and l_{scp} . During the inference, by enforcing the consistency of l_{scp} and l_o , l_{op} can be directly recovered, e.g. known $l_o = \text{horse}$ and $l_{scp} = \text{leg}_1$, then we get $l_{op} = \text{horse-leg}$. Currently we manually group those part labels. Nonetheless, automatically generating SCP is a very interesting problem especially with increased number of object categories, and will be investigated in the future.

5.2.2 Deep part and object potentials

In this section, we mainly describe the joint prediction of the semantic compositional parts (SCP) potentials and object potentials by a two-stream FCN, which is then used to construct the FCRF. Specifically, as illustrated in the framework (Fig. 5.2), the first channel of the FCN predicts a $(N_p + 1)$ -channel SCP potential map (N_p is the SCP label number, while the additional label represents the background). Similarly, the second channel of the network predicts a $(N_o + 1)$ -channel object-class potential map (N_o being the object class number). In addition, we concatenate the SCP potentials and the object potentials as a set of high-level features, and feed them to a new $k \times k$ conv. layer for object potential refinement. This predicted joint object potential has less noise

within the object and better boundaries. It is because firstly, the conv. layer accesses larger context for reducing the confusions, and secondly, our SCP potentials keep better edge information that can interact with the object potentials during joint training. As shown in Tab. 5.2, compared with the original FCN, the “Joint FCN” produce better performance.

One may consider to also generate refined SCP potentials similarly using the joint prediction layer. However, using the SCP potentials in this way does not show much improvement in our experiments. This is because firstly, the SCP potentials have already encoded more detailed boundary information than the object potentials, and the interactions would not help the SCP potentials to refine their boundaries. Secondly, the ambiguity of similar parts from different objects, which is the most challenging problem in part segmentation, has already been better addressed by using part sharing and the object-scale FCRF. Therefore, the joint prediction for SCP refinement is not adopted in our framework to reduce the system complexity.

Last but not the least, the SCP potentials and object potentials actually need different levels of context. This is also a key factor in DPM [FGM10], where they use a larger image scale for part filters and a smaller scale for root filters. In our case, we adopt a similar strategy with different input resolutions to obtain proper receptive fields, i.e. $s_p \times s_p$ for SCP and $s_o \times s_o$ for object, where we require $s_p > s_o$. We investigated the influence of input scale in our experiments, and chose the optimal s_p and s_o using cross-validation.

5.2.3 SCP segments proposal

As mentioned earlier, our FCRF is built upon a compact set of SCP segments. In our case, traditional object proposal algorithms such as CPMC [CCB12] or MCG [APB14] would typically fail due to the subtle difference of appearance between connected parts such as the leg and body. Nevertheless, we can use the SCP FCN network to generate accurate segments that are associated with SCP. Based on the predicted $(N_p + 1)$ -channel SCP probability map, we assign the SCP label with the highest probability to each pixel and generate the SCP label map. SCP segments are then generated by grouping the pixels with the same SCP labels.

Fig. 5.4 shows several examples of the proposed SCP segments from an image. We can see our

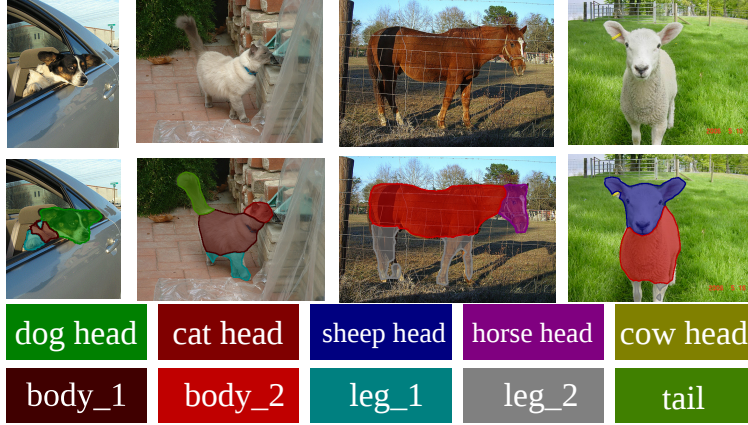


Figure 5.4: The SCP segments proposals handle various difficulties such as truncation, occlusion, deformation and view point changes. The colors maps shown at the bottom for each kind of SCP. We keep the color consistent in the results and will show more in our supplementary materials.

SCP segments work very well in terms of capturing correct semantic part regions and locating the object boundaries. To practically evaluate the segments, we did an oracle experiments by assigning the proposed SCP segments to be the overlapping ground truth class labels. The best possible object IOU is 85.1% over the quadrupeds animal set in Sec. 5.3.3, which performs reasonably well for capturing the object boundaries. One might also apply local dense pixel-wise refine strategy [CPK15] to further refine the segments, but it will increase computational cost and need further investigation.

5.2.4 Joint FCRF

FCN is essentially a sliding-window based approach, where its receptive field is always fixed to a local area given the input image, thus local confusion can hardly be avoided in many cases. Intuitively, the optimal receptive field should be close to the scale of the presented object in the image. Therefore, after generating SCP segments as well as SCP and object potentials from the joint FCN, we further construct a fully-connected CRF (FCRF) to automatically discover the object-scale context, where all the parts of an object can interact with each other, yielding the optimal semantic object and part prediction simultaneously.

Specifically, given the set of SCP segment proposals, we first cluster the proposed SCP segments into several groups according to their spatial distances, in case there are multiple isolated objects in an image. Two SCP segments are merged in the same group if the minimum distance of the pixels within these two segments is smaller than a threshold $t_s = 10$. We assume each group of SCP segments forms an object or overlapping objects in the image, which provides a good estimation of the object-scale. Then, we build one FCRF for each corresponding group. Formally, the FCRF can be represented as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of SCP segments in the same group and $\mathcal{E}$ is the set of edges connecting every pair of segments. As introduced in our framework (Fig. 5.2), for each SCP segment $\mathbf{P}$, we want to predict its semantic part label $l_{op}(\mathbf{P})$ (defined in Sec. 5.2.1), but can be reduced to separately inferring the object label $l_o(\mathbf{P})$ and SCP label $l_{scp}(\mathbf{P})$ by enforcing their label consistency. In the following we use l_{op}^i for $l_{op}(\mathbf{P}_i)$, l_p^i for $l_{scp}(\mathbf{P}_i)$ and l_o^i for $l_o(\mathbf{P}_i)$ for simplicity.

In sum, our FCRF is formulated as,

$$\begin{aligned} \min_{\mathcal{L}} \sum_{i \in \mathcal{V}} \psi_i(l_{op}^i) + \lambda_e \sum_{i,j \in \mathcal{V}, i \neq j} \psi_{i,j}(l_{op}^i, l_{op}^j) \\ \text{where, } \psi_i(l_{op}^i) = \eta(l_o^i, l_p^i)(\psi_i^o(l_o^i)) + \lambda_p \psi_i^p(l_p^i); \\ \psi_{i,j}(l_{op}^i, l_{op}^j) = \eta(l_o^i, l_p^i) \eta(l_o^j, l_p^j) \psi_{i,j}^{op}(l_o^i, l_o^j, l_p^i, l_p^j); \end{aligned} \quad (5.1)$$

where $\mathcal{L}$ is the label set of the proposed SCP segments, λ_e and λ_p are balancing parameters. $\psi_i^o(l_o^i) = \sum_{\mathbf{x}_j \in \mathbf{P}_i} -\log(P(l_o(\mathbf{x}_j)))$, is the sum of pixel-wise object potentials inside the SCP segment $\mathbf{P}_i$, and $\psi_i^p(l_p^i)$ is similarly defined with the SCP potentials. $\eta(l_o^i, l_p^i)$ is a constraint for part and object combination, which is 1 if l_o^i, l_p^i is a meaningful combination, i.e. a connection exists between l_o^i and l_p^i in the grammar as in Fig. 5.3, and set to be ∞ otherwise.

In order to learn the pairwise potentials, we train a two-layer fully-connected neural network, which takes the features from a pair of segments i and j as input, and predicts the probabilities of the four labels $l_o^i, l_o^j, l_p^i, l_p^j$. The ground truth labels for each training segment are the most dominant labels of the pixels within the segment, and we adopt the multinomial logistic loss for training.

For the pairwise features, we consider multiple semantically meaningful cues from the pairwise relations, i.e. $\mathbf{f}_{ij} = [\mathbf{f}_i^T, \mathbf{f}_j^T, \kappa_{ij}, \theta_{j|i}]^T$, where $\mathbf{f}_i = [\mathbf{f}_{oi}^T, \mathbf{f}_{pi}^T, \mathbf{f}_{ai}^T]^T$ is a segment self-descriptor, $\kappa_{ij} = [d_{ij}^a, d_{ij}^p]^T$ is a spatial metric of the segment pair. $\theta_{j|i}$ is the relative angle. We summarize the

features as follows, (1) $\mathbf{f}_{oi}^T$: the mean of pixel-wise object potentials. (2) $\mathbf{f}_{pi}^T$: the mean of pixel-wise SCP potentials. (3) $\mathbf{f}_{ai}$: the segment area, normalized by the discovered object area. (4) d_{ij}^a : the appearance geodesic distance from segment i to j , which accumulates the edge weights along the path from i to j on the image. (5) d_{ij}^p : the Euclidean distance between the center of the two segments normalized by the height and width of the object. (6) $\theta_{j|i}$: the relative angle of the center of the segments j with respect to the center of segment i . Specifically for d_{ij}^a , the edge weight between the two neighbouring segments is the sum of the edgemap [LSS12] values over their overlapping boundary.

After the model is learned, given the pairwise feature $\mathbf{f}_{ij}$, the potentials $\psi_{i,j}^{op}(l_o^i, l_o^j, l_p^i, l_p^j)$ is computed as the negative log-likelihood, i.e. $-\log(P(l_o^i)P(l_o^j)P(l_p^i)P(l_p^j))$, where the probabilities are from the neural network prediction.

For inference, since our graphical model have a very small number of nodes (less than 15 SCP segments in average), using the efficient LBP [MWJ13], our algorithm can converge very fast within 5 iterations.

5.2.5 Relation to the DPM structure

While we are dealing with semantic segmentation and parsing, our approach follows the spirit of DPM [FGM10] in object detection. In [GID15], Girshick et. al connected the DPM root filter and part filters with the convolutional filters from CNN, and the distance transform can be regarded as an additional pooling and geometry filtering step. Our work aims to solve the segmentation task here but has analogy with DPM, where the object prediction can be considered as the root filter and SCP potentials are our learned part filters. The difference is that rather than firing when the target is at the exact center of the detection window, our model fires at every location inside the target which produces more accurate location estimation. Regarding the part and root geometry, rather than explicitly modeling the geometry of root-part distance transform as in DPM, we implicitly model these spatial relationships using spatial distances and relative angles as spatial features and learn the pair-wise potentials in the fully-connected graphic model, which is more data-driven and generalizes better for handling variations.

For inference, DPM tries to find the probability of the part-root locations $\mathbf{p}$ given the object label l_o through sliding window, i.e. $\max_{\mathbf{p} \in \mathbf{I}} P(\mathbf{p}|l_o)$. In our case, sliding-window for part and object localization is realized by our FCN, based on which we can infer over the smaller label space i.e. $\max_{l_o \in \mathcal{L}} P(l_o|\mathbf{p})$, at the object-scale context.

5.3 Experiments

In this section, we provide all the experimental details, and evaluate our approach in terms of different experimental settings to demonstrate the advantage of our approach. Specifically, we conduct experiments on the Horse-Cow parsing dataset introduced in [WY15], the PASCAL Quadrupeds dataset from [CML14], and our PASCAL Part benchmark, over which extensive comparisons are performed.

5.3.1 Model training.

Both of our FCN models for SCP and object prediction are based on the 16-stride (16s) FCN, since there is trivial improvement (less than 1% as shown in our Tab. 5.2 and Tab.2 of [LSD15b]), while significant more time required to train a 8-stride model (8s).

Data augmentation. Effective data augmentation is the key to the success of FCN. Thus in our case, we did sufficient augmentation by first cropping out the connected object masks using a random generated bounding box around it. The size of the crop is specially 1.3 times larger than the object bounding box, which is a rough localization of the object for generalization. Then, each cropped image is resized into 300×300 , based on which we further augmented using the ideas from [EF15], i.e. we perform 4 additional random cropping at the size of 200×200 , flipping, changing the color intensity by a random scale in $[0.7, 1.3]$ with a probability of 0.4 and rotation in $[-5, +5]$ degree with a probability of 0.5. In average, each image is augmented up to around 25 training samples.

Optimization and step-wise training. We fine-tune all our models step-by-step from the publicly available VGG-net [LSD15b]. For training the SCP FCN, we first train a 32s FCN with the learning rate as 10^{-4} for the final $conv_{fc8}$ layer which is the layer predicting the SCP potentials, and 10^{-5} learning rate for the layers after $pool4$, while we fix the layers before $pool4$. Then for the 16s FCN, we start with training the $conv_{pool4}$ layer which is used for predicting SCP potentials from the $pool4$'s output. Then, we use the trained layers as initialization for the final 16s FCN training. We further fine-tune the $conv_{fc8}$ and $conv_{pool4}$ layer with a learning rate as 10^{-5} . For training the joint object FCN, we fix the SCP FCN and concatenate the SCP potentials with the 16s object potentials, over which another convolutional layer $conv_{jnt}$ (with a kernel size of 5) is used to predict the joint object potentials. For fine-tuning this model, we use 10^{-4} learning rate for the $conv_{jnt}$ layer, while 10^{-6} learning rate for the $conv_{fc8}$ and $conv_{pool4}$ of the object FCN. In all the cases, we keep the batch size as 32. For the other parameters, we refer to the ones given by [LSD15b].

For the two-layer neural network in training the pairwise potentials of the fully-connected CRF, we set 32 hidden nodes, and use the RELU for non-linearity with a dropout rate of 0.2 to regularize the model. We use a batch size of 10000 and set the learning rate to be 10^{-2} . In learning the pair-wise term, other than separately training SCP and object labels, i.e. l_{scp} and l_o , one may also consider output the semantic part label l_{op} which lies in the joint space of object and part. Nevertheless, we found it is harder to train due to that a lot more data are required for prediction in a high dimensional output space. Thus we chose to train separately.

In addition, we found it is important to balance different classes for part segmentation since some part can be very small, e.g. tails. Specifically, for both FCN and FCRF, we set the weight for each class in terms of the portion of samples in the class. This helps the network avoid overwhelming of other dominant classes.

All our neural networks are based on the caffe platform [JSD14] and partially from the code provided by [LSD15b].

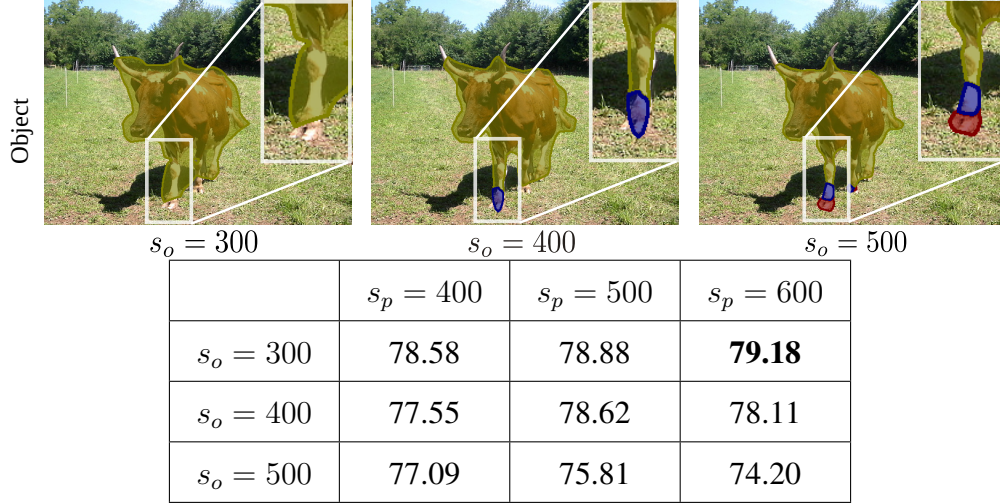


Figure 5.5: Investigation of image scales. Top: an example from our joint object FCN prediction, showing that larger scale leads to finer boundaries, but introduces local ambiguities. Bottom: validate the accuracy of scale combination of s_o and s_p .

5.3.2 Parameters and details

In the FCRF, we set $\lambda_e = 2$ and $\lambda_p = 0.3$ which are validated over a validation set from the Quadrupeds dataset. The same set is used for all other validation experiments. For inference over the graphical model, we use the LBP tool provided by Meltzer¹.

Investigation of input image scale. The input image scale for FCN is one of the most important factors for achieving good performance as also shown in recent works [CPK15, LSR15]. Fig. 5.5 shows the investigation of changing the input image scale for SCP proposals and object potentials. In the top row, we can see the larger the image scale is, the more accurate the object boundaries could be, while the more local confusion it has, *e.g.* a leg of the cow is starting to be confused with horse. Thus, we validate the scale of $s_p \in \{400, 500, 600\}$ and $s_o \in \{300, 400, 500\}$. At the bottom of Fig. 5.5, we show the validated results, and the optimal combination, i.e. $s_p = 600, s_o = 300$ is used in all our experiments.

¹<http://www.cs.huji.ac.il/talyam/inference.html>

Training and inference time. For training, we found the model would converge after 40k iterations and it takes around 2 days for SCP FCN, 1 day for object potentials and 8 hours for the graphical model with a platform of 4 core 3.2Hz CPU and a K40 GPU. For inference, in average, one image takes 0.3s for FCN forward propagation with the GPU and 1.3s for the graph model with our CPU.

5.3.3 Performance comparisons

We compare our algorithm with three state-of-the-art methods on the two tasks. For object segmentation, we compare our method with the FCN [LSD15b]. We use the code provided by the author and tune their model based on our dataset, including tuning a fixed optimal image scale for a fair comparison. For semantic part segmentation, we compare with the most recent compositional-based semantic part segmentation (SPS) [WY15] over the Horse-Cow parsing dataset, and the hypercolumn (HC) [HAG15] in all datasets. For comparing with HC [HAG15], as there is no available code or results from the author, we implemented a similar part parsing framework by first performing figure-ground mask with the trained FCN 8s [LSD15b], and then assign part labels inside with a optimally tuned image scale to be our baseline method. This baseline is close to HC and also performs well in the experiments, but is different from [HAG15] in some details, thus we mark it as HC*. For evaluation, we adopt the standard intersection over union (IOU) criteria.

Horse-Cow parsing dataset. The Horse-Cow dataset is a two-animal part segmentation benchmark proposed in [WY15]. For each animal class, they manually select the mostly observable animal instances in both train-val and test set from the PASCAL VOC 2010 [EVW10]. There are 294 training examples and 277 test images. Since it is not published yet, we asked the author for this dataset and their results. In [WY15], the task is to segment the part given the object class. Thus for a fair comparison, we also test every instance with the object class known. In such a case, our graphical model in Eqn.(5.1) is reduced to inferring the part label l_p with a binary object potential l_o .

Tab. 5.1 provides the results of SPS [WY15], HC* [HAG15] and ours over semantic part and figure-ground segmentation. Our method outperforms the previous state-of-the-art with a significant

margin, averagely 13% better than SPS and 4.5% better than HC*. Several qualitative examples are shown in Fig. 5.6, where we keep the color map consistent with the SPC labels (Fig. 5.4). In these cases, we can see that, built on explicit geometry rules, SPS is limited in handling large variance of object parts, which makes it difficult to model highly deformable cases like the tail, or occluded animals. For HC*, due to inaccuracy from the object mask, it may miss some detailed regions like legs such as the 1_{st} column of horse in Fig. 5.6.

Horse								
	Bkg	head	body	leg	tail	Fg	IOU	Pix. Acc
SPS [WY15]	79.14	47.64	69.74	38.85	-	68.63	-	81.45
HC* [HAG15]	85.71	57.30	77.88	51.93	37.10	78.84	61.98	87.18
Ours	87.34	60.02	77.52	58.35	51.88	80.70	65.02	88.49

Cow								
	Bkg	head	body	leg	tail	Fg	IOU	Pix. Acc
SPS [WY15]	78.00	40.55	61.65	36.32	-	71.98	-	78.97
HC* [HAG15]	81.86	55.18	72.75	42.03	11.04	77.07	52.57	84.43
Ours	85.68	58.04	76.04	51.12	15.00	82.63	57.18	87.00

Table 5.1: Average precision over the Horse-Cow dataset.

Quadrupeds dataset. We further extend our experiments into the Quadrupeds part dataset which contains five animal classes, i.e., cat, dog, sheep, cow and horse. In this task, we simultaneously predict the object and part masks. We obtain the data given by [CML14], which include all the part labels in PASCAL VOC 2010 training and validation images. We select the images containing the target objects, and treat the validation set as our test images, resulting in 3120 training images and 294 testing images. In addition, since we are focusing more on semantic part segmentation, during testing, we roughly localize the object inside by using the strategy in data augmentation (Sec. 5.3.1). We will provide our code of this localization to others for a fair comparison. It should be noted



Figure 5.6: Comparison examples with SPS [WY15] and HC [HAG15] in the Horse-Cow dataset and the color map is in Fig. 5.4 (Best view in color).

that such localization is still very coarse with much looser bounding boxes than the ones needed in previous parsing methods [DCY14, WY15].

Object segmentation accuracy								
	Bkg	Dog	Cat	Cow	Horse	Sheep	IOU	Pix. Acc
FCN 16s [LSD15b]	93.25	74.30	78.62	61.88	56.56	67.63	72.04	93.00
FCN 8s [LSD15b]	93.55	74.39	78.52	60.81	58.39	69.15	72.47	93.17
Joint FCN(16s)	94.04	75.13	80.52	66.76	63.04	71.54	75.17	93.77
FCRF+FCN(16s)	93.88	77.10	80.92	68.76	63.40	64.54	74.57	93.87
Ours (w/o SCP)	93.29	77.50	77.81	70.24	68.94	71.31	76.51	93.89
Ours final	94.40	79.03	83.04	74.82	69.94	70.59	78.64	94.71

Semantic part segmentation accuracy								
	Bkg	Dog	Cat	Cow	Horse	Sheep	IOU	Pix. Acc
HC* [HAG15]	92.83	42.07	43.99	35.49	38.59	33.80	41.36	89.54
Ours (w/o SCP)	93.31	40.71	43.37	36.65	43.65	32.81	42.01	90.68
Ours final	94.46	45.63	47.81	42.7	49.60	35.74	46.69	91.74

Table 5.2: Average precision over the Quadrupeds data.

Following HC, the parts of all the quadrupeds are labelled into head, body, leg and tail, from which we construct a shared SCP grammar as introduced in Sec. 5.2.1. In our grammar, horse, cow and sheep share the same body and leg type, while cat and dog share others. All the animals share a tail label, and each animal has its own head label since the head is highly distinguishable [PVJ11]. In total, 10 SCPs are used, while 20 labels are need if all the parts are treated independently.

Tab. 5.2 shows the compared results on both object and part segmentation. As shown, in terms of object segmentation, comparing with FCN [LSD15b], our final results improves 6.2%, which demonstrates the advantage of the joint model. We also compare the variants of our approach with different components. “Joint FCN(16s)” produces the object masks directly from the two-stream FCN without FCRF inference. By including the SCP potentials, the object prediction of “Joint FCN



Figure 5.7: Comparison examples with FCN [LSD15b] for object segmentation and HC* [HAG15] for part segmentation in the Quadrupeds dataset. The object label color map is shown at above and part label color map is shown in Fig. 5.4 (Best view in color).

(16s)” already out-performs FCN, which shows that the joint object potentials have less pixel-wise confusion. In addition, as shown in the “FCRF+FCN(16s)”, using the FCRF with the FCN object potentials without the joint convolutional layer improves over 2% compared to “FCN(16s)”, showing FCRF inference can help refine segmentation results by exploring long-range context. Moreover, with the joint potentials, the performance of our full model has another 4% boost. This shows the joint FCN object potentials provide better evidence for our graphical model and is essential to our system. At last, in “Ours (w/o SCP)”, we investigate the usefulness of SCP by not allowing the part sharing. The results drops 2% and 4% for object and part respectively compared to our final results. This is mostly because more confusion happened for parts with similar appearance, yielding worse part segments.

We show several qualitative comparison examples with the FCN at left of Fig. 5.7, in which our algorithm is able to solve the local ambiguities that FCN usually encounters. For instance, at the 4_{th} row, the legs of a cow are confused with horse using FCN. In contrast, by borrowing the object-scale evidence from the cow body and cow head, our model can correct this local confusion. In addition, as shown in the horse segments at the 3_{rd} row, our SCP segments are able to provide more precise object boundaries in many difficult cases, like the legs of the horse crossing the bar.

For semantic part segmentation, we summarized the mean IOU of all parts for each object in Tab. 5.2 due to space limitation. We will include more detailed results in our supplementary materials. Our results are also significantly higher than the results of HC* [HAG15] (over 5%). As HC performs object and part segmentation sequentially, the errors in object predictions, including local confusion and inaccurate boundaries, will propagate to the parts. For example, at the 2_{nd} row of Fig. 5.7, the cat and dog are confused in the object FCN prediction, which makes part of the cat head errorly labelled as dog head in HC. We solve such problems through the FCRF by considering object-scale context. In addition, thanks to our optimized image scale for both object and part, our method can capture part boundaries that are sometimes missed by the FCN and HC. We provide more results in our supplementary materials.

PASCAL part segmentation benchmark. In addition to the labels from the train-validation set of [CML14], following the same hierarchical part labelling system, we additionally labelled

semantic parts over the PASCAL VOC 2010 test set [EVWa] of the object segmentation task, which includes 994 images. With respect to the PASCAL VOC test benchmark, we are not going to release the labels and will instead launch an evaluation server for researchers to fairly compare their part segmentation results.

We test our algorithm over images with the five quadrupeds, which include 281 images. As shown in Tab. 5.3, the results are consistent with that from the validation set, and our method achieves the best overall IOU outperforming the state-of-the-art with a large margin.

Object segmentation accuracy								
	Bkg	Dog	Cat	Cow	Horse	Sheep	IOU	Pix. Acc
FCN 8s [LSD15b]	94.45	70.14	75.45	64.06	64.75	69.06	72.99	93.90
Ours final	95.31	77.44	80.47	72.13	76.18	67.96	78.25	95.26

Semantic part segmentation accuracy								
	Bkg	Dog	Cat	Cow	Horse	Sheep	IOU	Pix. Acc
HC* [HAG15]	94.36	41.24	42.42	35.22	45.00	38.86	43.11	90.64
Ours final	95.14	46.52	48.06	41.80	56.67	36.02	48.16	92.47

Table 5.3: Average precision over the part segmentation benchmark.

5.4 Conclusion and discussion

In this paper, we proposed the framework for jointly solving object and part segmentation. Our approach follows the spirit of DPM, and leverages the advantages of both sides, yielding the state-of-the-art results. Recently, for object segmentation, there are other methods that provides additional improvement over the FCN such as adding pixel-wise Dense CRF [CPK15, LSR15]. Our framework is complementary to them in terms of modelling over parts and adaptive to object scale for solving local ambiguity. In addition, we tackles the part segmentation beyond the object. Possible failure cases for us would be strong appearance confusion, strong occlusion, where object

scale or SCP segments can be misled, yielding inaccurate results (*e.g.* the back leg of the dog in the first example of Fig. 5.7).

In the future, we will try to automatically learn the SCP, and jointly model the instance segmentation to incorporate detection, which provides better object scale and solve the remaining localization issue.

CHAPTER 6

Discussion, conclusion and future directions

In this thesis, we majorly show how to perform joint multiple task learning and inference with deep networks, especially fully convolutional network (FCN), and conditional random fields (CRF) using three cases, i.e. occlusion boundary estimation, joint segmentation and geometry estimation, and hierarchical object segmentation. In all the cases, we show joint inference results are better in terms of both visual qualities and quantitative evaluation. In addition, we also extend the popular PASCAL semantic part and instance dataset with occlusion boundary between different object instances, yielding a more detailed labeled dataset for multi-tasks analysis.

However, although we observe improvement through joint multiple task learning and inference, there still exists many failure cases in our experiments. For example, when the inferred results from some of the learned network are dramatically wrong (which is usually feed as unary), the error might propagate through the following CRF. In addition, since the complexity of including many different tasks are growing in a polynomial order when building connections between them, yielding a vast requirements of well labelled data in practice which is hard to collect. Though building synthetic system similar with SceneNet [MHL16], unreal CV [QY16] and shapenet [CFG15] etc. are one of the choice, but simulating various scene and lighting conditions are still difficult. Thus, there is still a very long way towards solving of computer vision.

Promisingly, there are still some light at the front of solving those difficulties as mentioned above through unsupervised learning especially through videos. As we know, dramatic information is existing in motion, optical flow, structure from motion or disparity through low-level pixel matching. Based on video, we may automatically generate many information like occlusion boundary, 3D surface and geometry, temporal cues through tracking, which could be used to learn deep models performing on a single image. Reversely, the model learned on single image can compensate some

missing matching in video during the information extraction. Thus by unifying the learning and inference from a single image and a video stream are one of the most promising way of handling the difficulty of acquire vast number of training data with labelled ground truth to learn single image model. We believe in the near future, video from nature images (collecting from website like Youtube or Youku) will be considered and used as a major learning resource for single image models, especially 3D reconstruction models.

REFERENCES

- [ABD10] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. “Fast High-Dimensional Filtering Using the Permutohedral Lattice.” In *Computer Graphics Forum*, volume 29, pp. 753–762. Wiley Online Library, 2010.
- [AHG12] Pablo Arbelaez, Bharath Hariharan, Chunhui Gu, Saurabh Gupta, Lubomir D. Bourdev, and Jitendra Malik. “Semantic segmentation using regions and parts.” In *CVPR*, pp. 3378–3385, 2012.
- [AL12] Hossein Azizpour and Ivan Laptev. “Object Detection Using Strongly-Supervised Deformable Part Models.” In *ECCV*, pp. 836–849, 2012.
- [AMF11] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. “Contour Detection and Hierarchical Image Segmentation.” *IEEE Trans. Pattern Anal. Mach. Intell.*, **33**(5):898–916, 2011.
- [APB14] Pablo Andrés Arbeláez, Jordi Pont-Tuset, Jonathan T. Barron, Ferran Marqués, and Jitendra Malik. “Multiscale Combinatorial Grouping.” In *CVPR*, pp. 328–335, 2014.
- [ARS12] Alper Ayvaci, Michalis Raptis, and Stefano Soatto. “Sparse occlusion detection with optical flow.” *International Journal of Computer Vision*, **97**(3):322–338, 2012.
- [BAS15] Jonathan T Barron, Andrew Adams, YiChang Shih, and Carlos Hernández. “Fast Bilateral-Space Stereo for Synthetic Defocus.” *CVPR*, 2015.
- [BF11] Yihang Bo and Charless C. Fowlkes. “Shape-based pedestrian parsing.” In *CVPR*, pp. 2265–2272, 2011.
- [Bie81] Irving Biederman. “On the semantics of a glance at a scene.” In *In M. Kubovy and J. R. Pomerantz (Eds.), Perceptual organization*, pp. 213–263, 1981.
- [BM92] Peter N Belhumeur and David Mumford. “A Bayesian treatment of the stereo correspondence problem using half-occluded regions.” In *CVPR*, pp. 506–512, 1992.
- [BM09a] Lubomir Bourdev and Jitendra Malik. “Poselets: Body part detectors trained using 3d human pose annotations.” In *2009 IEEE 12th International Conference on Computer Vision*, pp. 1365–1372. IEEE, 2009.
- [BM09b] Lubomir D. Bourdev and Jitendra Malik. “Poselets: Body part detectors trained using 3D human pose annotations.” In *ICCV*, pp. 1365–1372, 2009.
- [BP15] Jonathan T. Barron and Ben Poole. “The Fast Bilateral Solver.” *CoRR*, 2015.
- [BRG16] Aayush Bansal, Bryan Russell, and Abhinav Gupta. “Marr Revisited: 2D-3D Alignment via Surface Normal Prediction.” In *CVPR*, 2016.

- [BST16] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. “Semantic Segmentation With Boundary Neural Fields.” In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [CCB12] João Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. “Semantic Segmentation with Second-Order Pooling.” In *ECCV (7)*, pp. 430–443, 2012.
- [CFG15] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. “ShapeNet: An Information-Rich 3D Model Repository.” *CoRR*, **abs/1512.03012**, 2015.
- [CML14] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan L. Yuille. “Detect What You Can: Detecting and Representing Objects Using Holistic Models and Body Parts.” In *CVPR*, pp. 1979–1986, 2014.
- [Coo97] Martin C. Cooper. “Interpreting line drawings of curved objects with tangential edges and surfaces.” *Image Vision Comput.*, **15**(4):263–276, 1997.
- [CPK15] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. “Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs.” In *ICLR*, 2015.
- [CSN07] Edward Craft, Hartmut Schütze, Ernst Niebur, and Rüdiger Von Der Heydt. “A neural model of figure–ground organization.” *Journal of neurophysiology*, **97**(6):4310–4326, 2007.
- [CSS09] Bryan C. Catanzaro, Bor-Yiing Su, Narayanan Sundaram, Yunsup Lee, Mark Murphy, and Kurt Keutzer. “Efficient, high-quality image contour detection.” In *ICCV*, pp. 2381–2388, 2009.
- [DCS14] Jian Dong, Qiang Chen, Xiaohui Shen, Jianchao Yang, and Shuicheng Yan. “Towards Unified Human Parsing and Pose Estimation.” In *CVPR*, 2014.
- [DCY14] Jian Dong, Qiang Chen, Shuicheng Yan, and Alan Yuille. “Towards unified object detection and semantic segmentation.” In *European Conference on Computer Vision*, pp. 299–314. Springer, 2014.
- [DDS09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. “ImageNet: A large-scale hierarchical image database.” In *CVPR*, pp. 248–255, 2009.
- [DGE14] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. “Context as Supervisory Signal: Discovering Objects with Predictable Context.” In *ECCV*, pp. 362–377, 2014.
- [DHH09] Santosh Kumar Divvala, Derek Hoiem, James Hays, Alexei A. Efros, and Martial Hebert. “An empirical study of context in object detection.” In *CVPR*, pp. 1271–1278, 2009.
- [DT05] Navneet Dalal and Bill Triggs. “Histograms of Oriented Gradients for Human Detection.” In *CVPR*, 2005.

- [DZ15] Piotr Dollár and C. Lawrence Zitnick. “Fast Edge Detection Using Structured Forests.” *IEEE Trans. Pattern Anal. Mach. Intell.*, **37**(8):1558–1570, 2015.
- [EF15] David Eigen and Rob Fergus. “Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-scale Convolutional Architecture.” In *ICCV*, 2015.
- [EPF14] David Eigen, Christian Puhrsch, and Rob Fergus. “Depth Map Prediction from a Single Image using a Multi-Scale Deep Network.” In *NIPS*. 2014.
- [EVWa] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. “The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results.” <http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html>.
- [EVWb] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. “The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.” <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>.
- [EVW10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. “The Pascal Visual Object Classes (VOC) Challenge.” *IJCV*, **88**(2):303–338, 2010.
- [EW12] S. M. Ali Eslami and Christopher K. I. Williams. “A Generative Model for Parts-based Object Segmentation.” In *NIPS*, pp. 100–107, 2012.
- [FBK09] Fang Fang, Huseyin Boyaci, and Daniel Kersten. “Border ownership selectivity in human early visual cortex and its modulation by attention.” *The Journal of Neuroscience*, **29**(2):460–465, 2009.
- [FGM10] Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. “Object Detection with Discriminatively Trained Part-Based Models.” *IEEE Trans. Pattern Anal. Mach. Intell.*, **32**(9):1627–1645, 2010.
- [GAL12] Chunhui Gu, Pablo Andrés Arbeláez, Yuanqing Lin, Kai Yu, and Jitendra Malik. “Multi-component Models for Object Detection.” In *ECCV*, pp. 445–458, 2012.
- [GDD14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation.” June 2014.
- [GFM11] Ross B. Girshick, Pedro F. Felzenszwalb, and David A. McAllester. “Object Detection with Grammar Models.” In *NIPS*, pp. 442–450, 2011.
- [GGA14] Saurabh Gupta, Ross Girshick, Pablo Arbelaez, and Jitendra Malik. “Learning Rich Features from RGB-D Images for Object Detection and Segmentation.” In *ECCV*. 2014.
- [GH13] Ruiqi Guo and Derek Hoiem. “Support Surface Prediction in Indoor Scenes.” In *ICCV*, 2013.
- [Gib68] J Gibson. “The perception of surface layout: A classification of types.” *Unpublished AIJ Purple Perils essay*, 1968.

- [GID15] Ross Girshick, Forrest Iandola, Trevor Darrell, and Jitendra Malik. “Deformable Part Models are Convolutional Neural Networks.” In *CVPR*, 2015.
- [GLY92] Davi Geiger, Bruce Ladendorf, and Alan Yuille. “Occlusions and binocular stereo.” In *Computer Vision—ECCV’92*, pp. 425–433. Springer, 1992.
- [HAB11] Bharath Hariharan, Pablo Arbelaez, Lubomir D. Bourdev, Subhransu Maji, and Jitendra Malik. “Semantic contours from inverse detectors.” In *ICCV*, pp. 991–998, 2011.
- [HAG14] Bharath Hariharan, Pablo Andrés Arbeláez, Ross B. Girshick, and Jitendra Malik. “Simultaneous Detection and Segmentation.” In *ECCV*, pp. 297–312, 2014.
- [HAG15] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. “Hypercolumns for Object Segmentation and Fine-grained Localization.” In *CVPR*, 2015.
- [HEH07] Derek Hoiem, Alexei A. Efros, and Martial Hebert. “Recovering Surface Layout from an Image.” *IJCV*, **75**(1):151–172, 2007.
- [HEH11] Derek Hoiem, Alexei A. Efros, and Martial Hebert. “Recovering Occlusion Boundaries from an Image.” *IJCV*, **91**(3):328–346, 2011.
- [HMK15] Katrin Honauer, Lena Maier-Hein, and Daniel Kondermann. “The HCI Stereo Metrics: Geometry-Aware Performance Analysis of Stereo Algorithms.” In *ICCV*, 2015.
- [HMQ05] Rüdiger von der Heydt, Todd Macuda, and Fangtu T Qiu. “Border-ownership-dependent tilt aftereffect.” *JOSA A*, **22**(10):2222–2229, 2005.
- [HY10] Xuming He and Alan L. Yuille. “Occlusion Boundary Detection Using Pseudo-depth.” In *ECCV*, pp. 539–552, 2010.
- [HYK13] Xiaodi Hou, Alan Yuille, and Christof Koch. “Boundary Detection Benchmarking: Beyond F-Measures.” In *CVPR*, pp. 2123–2130, 2013.
- [HZR16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- [IYF15] Satoshi Ikehata, Hang Yang, and Yasutaka Furukawa. “Structured Indoor Modeling.” In *ICCV*, 2015.
- [JSD14] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. “Caffe: Convolutional Architecture for Fast Feature Embedding.” *arXiv preprint arXiv:1408.5093*, 2014.
- [KG76] Gaetano Kanizsa and Walter Gerbino. “Convexity and symmetry in figure-ground organization.” *Vision and artifact*, pp. 25–32, 1976.
- [KK12a] Philipp Krähenbühl and Vladlen Koltun. “Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials.” *NIPS*, 2012.

- [KK12b] Philipp Krähenbühl and Vladlen Koltun. “Efficient Nonlocal Regularization for Optical Flow.” In *ECCV*, 2012.
- [KK13] Philipp Krähenbühl and Vladlen Koltun. “Parameter Learning and Convergent Inference for Dense Random Fields.” In *ICML*, 2013.
- [KLD14] Abhijit Kundu, Yin Li, Frank Daellert, Fuxin Li, and James M. Rehg. “Joint Semantic Segmentation and 3D Reconstruction from Monocular Video.” In *ECCV*, 2014.
- [Kok16a] Iasonas Kokkinos. “Pushing the Boundaries of Boundary Detection using Deep Learning.” *ICLR*, 2016.
- [Kok16b] Iasonas Kokkinos. “UberNet: Training a Universal Convolutional Neural Network for Low-, Mid-, and High-Level Vision using Diverse Datasets and Limited Memory.” *arXiv preprint arXiv:1609.02132*, 2016.
- [KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks.” In *NIPS*, pp. 1106–1114, 2012.
- [LGK10] Beyang Liu, Stephen Gould, and Daphne Koller. “Single image depth estimation from predicted semantic labels.” In *CVPR*, pp. 1253–1260, 2010.
- [LL09] Ido Leichter and Michael Lindenbaum. “Boundary ownership by lifting to 2.1D.” In *ICCV*, pp. 9–16, 2009.
- [LLY14] Wenhao Lu, Xiaochen Lian, and Alan L. Yuille. “Parsing Semantic Parts of Cars Using Graphical Models and Segment Appearance Consistency.” In *BMVC*, 2014.
- [LMP01] John Lafferty, Andrew McCallum, Fernando Pereira, et al. “Conditional random fields: Probabilistic models for segmenting and labeling sequence data.” In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pp. 282–289, 2001.
- [Low04] DG Lowe. “Distinctive image features from scale-invariant keypoints.” *IJCV*, 2004.
- [LRB15] Wei Liu, Andrew Rabinovich, and Alexander C. Berg. “ParseNet: Looking Wider to See Better.” *CoRR*, **abs/1506.04579**, 2015.
- [LSD15a] Bo Li, Chunhua Shen, Yuchao Dai, Anton van den Hengel, and Mingyi He. “Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs.” In *CVPR*, pp. 1119–1127, 2015.
- [LSD15b] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully Convolutional Networks for Semantic Segmentation.” In *CVPR*, pp. 3431–3440, 2015.
- [LSL15a] Fayao Liu, Chunhua Shen, and Guosheng Lin. “Deep Convolutional Neural Fields for Depth Estimation From a Single Image.” In *CVPR*, June 2015.

- [LSL15b] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian D. Reid. “Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields.” *CoRR*, **abs/1502.07411**, 2015.
- [LSR15] Guosheng Lin, Chunhua Shen, Ian Reid, and Anton van den Hengel. “Efficient piecewise training of deep structured models for semantic segmentation.”, 2015.
- [LSS12] Marius Leordeanu, Rahul Sukthankar, and Cristian Sminchisescu. “Efficient Closed-Form Solution to Generalized Boundary Detection.” In *ECCV (4)*, pp. 516–529, 2012.
- [LTH16] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network.” *CoRR*, **abs/1609.04802**, 2016.
- [LZP14] Lubor Ladicky, Bernhard Zeisl, and Marc Pollefeys. “Discriminatively Trained Dense Surface Normal Estimation.” In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *ECCV*, 2014.
- [MFM04] David R. Martin, Charless Fowlkes, and Jitendra Malik. “Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues.” *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**(5):530–549, 2004.
- [MHL16] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J Davison. “SceneNet RGB-D: 5M Photorealistic Images of Synthetic Indoor Trajectories with Ground Truth.” *arXiv preprint arXiv:1612.05079*, 2016.
- [MV15] Aravindh Mahendran and Andrea Vedaldi. “Understanding deep image representations by inverting them.” In *CVPR*, pp. 5188–5196, 2015.
- [MWJ13] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. “Loopy Belief Propagation for Approximate Inference: An Empirical Study.” In *CoRR*, 2013.
- [MYP11] Michael Maire, Stella X. Yu, and Pietro Perona. “Object detection and segmentation from joint embedding of parts and pixels.” In *ICCV*, pp. 2142–2149, 2011.
- [NHH15] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. “Learning deconvolution network for semantic segmentation.” In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1520–1528, 2015.
- [NM90] Mark Nitzberg and David Mumford. “The 2.1-D sketch.” In *ICCV*, pp. 138–144, 1990.
- [NYD16] Alejandro Newell, Kaiyu Yang, and Jia Deng. “Stacked hourglass networks for human pose estimation.” In *European Conference on Computer Vision*, pp. 483–499. Springer, 2016.
- [OE11] Patrick Ott and Mark Everingham. “Shared parts for deformable part-based models.” In *CVPR*, pp. 1513–1520, 2011.

- [PG08] Stephen E Palmer and Tandra Ghose. “Extremal Edge A Powerful Cue to Depth Perception and Figure-Ground Organization.” *Psychological Science*, **19**(1):77–83, 2008.
- [PVJ11] Omkar M. Parkhi, Andrea Vedaldi, C. V. Jawahar, and Andrew Zisserman. “The truth about cats and dogs.” In *ICCV*, pp. 1427–1434, 2011.
- [QY16] Weichao Qiu and Alan Yuille. “UnrealCV: Connecting Computer Vision to Unreal Engine.” In *Computer Vision–ECCV 2016 Workshops*, pp. 909–916. Springer, 2016.
- [RFM06] Xiaofeng Ren, Charless Fowlkes, and Jitendra Malik. “Figure/Ground Assignment in Natural Images.” In *ECCV*, pp. 614–627, 2006.
- [Rob63] Lawrence G. Roberts. *Machine Perception of Three-Dimensional Solids*. Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York, 1963.
- [Sau05] Eric Saund. “Logic and MRF Circuitry for Labeling Occluding and Thinline Visual Contours.” In *NIPS*, pp. 1153–1159, 2005.
- [SBM11] Patrik Sundberg, Thomas Brox, Michael Maire, Pablo Arbelaez, and Jitendra Malik. “Occlusion boundary detection and figure/ground assignment from optical flow.” In *CVPR*, pp. 2233–2240, 2011.
- [SFP13] Alexander G. Schwing, Sanja Fidler, Marc Pollefeys, and Raquel Urtasun. “Box in the Box: Joint 3D Layout and Object Reasoning from Single Images.” In *ICCV*, pp. 353–360. IEEE Computer Society, 2013.
- [SH09] Andrew N. Stein and Martial Hebert. “Occlusion Boundaries from Motion: Low-Level Detection and Mid-Level Reasoning.” *IJCV*, **82**(3):325–357, 2009.
- [SHK12] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. “Indoor Segmentation and Support Inference from RGBD Images.” In *ECCV*, pp. 746–760, 2012.
- [SHK14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. “Dropout: A simple way to prevent neural networks from overfitting.” *The Journal of Machine Learning Research*, **15**(1), 2014.
- [SLX15] Shuran Song, Samuel Lichtenberg, and Jianxiong Xiao. “SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite.” In *CVPR*, 2015.
- [SM00] Jianbo Shi and Jitendra Malik. “Normalized Cuts and Image Segmentation.” *PAMI*, **22**(8):888–905, 2000.
- [SS11] Min Sun and Silvio Savarese. “Articulated part-based model for joint object detection and pose estimation.” In *ICCV*, pp. 723–730, 2011.
- [SSJ16] Laura Sevilla-Lara, Deqing Sun, Varun Jampani, and Michael J. Black. “Optical Flow With Semantic Segmentation and Localized Layers.” In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

- [SSN09] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. “Make3D: Learning 3D Scene Structure from a Single Still Image.” *IEEE Trans. Pattern Anal. Mach. Intell.*, **31**(5):824–840, 2009.
- [SSP14] Filip Srajer, Alexander G. Schwing, Marc Pollefeys, and Tomas Pajdla. “Match Box: Indoor Image Matching via Box-Like Scene Estimation.” In *2nd International Conference on 3D Vision, 3DV 2014, Tokyo, Japan, December 8-11, 2014, Volume 1*, 2014.
- [Sug84] Kokichi Sugihara. “An algebraic approach to shape-from-image problems.” *Artificial intelligence*, **23**(1):59–95, 1984.
- [SZ14] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” *CoRR*, **abs/1409.1556**, 2014.
- [TB10] Zhuowen Tu and Xiang Bai. “Auto-Context and Its Application to High-Level Vision Tasks and 3D Brain Image Segmentation.” *IEEE Trans. Pattern Anal. Mach. Intell.*, **32**(10):1744–1757, 2010.
- [TCY05] Zhuowen Tu, Xiangrong Chen, Alan L. Yuille, and Song Chun Zhu. “Image Parsing: Unifying Segmentation, Detection, and Recognition.” *International Journal of Computer Vision*, **63**(2):113–140, 2005.
- [TFA15] Ching Lik Teo, Cornelia Fermuller, and Yiannis Aloimonos. “Fast 2D border ownership assignment.” In *CVPR*, pp. 5117–5125, 2015.
- [TFL08] Alexander Thomas, Vittorio Ferrari, Bastian Leibe, Tinne Tuytelaars, and Luc J. Van Gool. “Using Recognition to Guide a Robot’s Attention.” In *RSS*, 2008.
- [TS14] Alexander Toshev and Christian Szegedy. “DeepPose: Human Pose Estimation via Deep Neural Networks.” In *CVPR*, pp. 1653–1660, 2014.
- [WEF14] Li Wan, David Eigen, and Rob Fergus. “End-to-End Integration of a Convolutional Network, Deformable Parts Model and Non-Maximum Suppression.” *arXiv preprint arXiv:1411.5309*, 2014.
- [WFG15] Xiaolong Wang, David Fouhey, and Abhinav Gupta. “Designing Deep Networks for Surface Normal Estimation.” In *CVPR*, 2015.
- [WRH15] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. “Learning to detect Motion Boundaries.” In *CVPR*, pp. 2578–2586, 2015.
- [WSL15a] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian L. Price, and Alan L. Yuille. “Towards unified depth and semantic prediction from a single image.” In *CVPR*, pp. 2800–2809, 2015.
- [WSL15b] Peng Wang, Xiaohui Shen, Zhe L. Lin, Scott Cohen, Brian L. Price, and Alan L. Yuille. “Joint Object and Part Segmentation using Deep Learned Potentials.” *ICCV*, 2015.

- [WSR16] Peng Wang, Xiaohui Shen, Bryan Russell, Scott Cohen, Brian Price, and Alan L Yuille. “SURGE: Surface Regularized Geometry Estimation from a Single Image.” In *Advances in Neural Information Processing Systems*, pp. 172–180, 2016.
- [WY15] Jianyu Wang and Alan Yuille. “Semantic Part Segmentation using Compositional Model combining Shape and Appearance.” In *CVPR*, 2015.
- [WY16] Peng Wang and Alan Yuille. “DOC: Deep OCclusion Estimation from a Single Image.” In *European Conference on Computer Vision*, pp. 545–561. Springer, 2016.
- [XF12] Jianxiong Xiao and Yasutaka Furukawa. “Reconstructing the World’s Museums.” In *ECCV*, 2012.
- [XRL14] Li Xu, Jimmy SJ Ren, Ce Liu, and Jiaya Jia. “Deep convolutional neural network for image deconvolution.” In *Advances in Neural Information Processing Systems*, pp. 1790–1798, 2014.
- [XT15] Saining Xie and Zhuowen Tu. “Holistically-Nested Edge Detection.” *ICCV*, 2015.
- [XXC12] Junyuan Xie, Linli Xu, and Enhong Chen. “Image denoising and inpainting with deep neural networks.” In *Advances in Neural Information Processing Systems*, pp. 341–349, 2012.
- [YFU12] Jian Yao, Sanja Fidler, and Raquel Urtasun. “Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation.” In *CVPR*, pp. 702–709, 2012.
- [YKO12] Kota Yamaguchi, M. Hadi Kiapour, Luis E. Ortiz, and Tamara L. Berg. “Parsing clothing in fashion photographs.” In *CVPR*, pp. 3570–3577, 2012.
- [YLK01] Stella X. Yu, Tai Sing Lee, and Takeo Kanade. “A Hierarchical Markov Random Field Model for Figure-Ground Segregation.” In *EMMCVPR*, pp. 118–133, 2001.
- [YR11] Yi Yang and Deva Ramanan. “Articulated pose estimation with flexible mixtures-of-parts.” In *CVPR*, pp. 1385–1392, 2011.
- [ZCL11] Long Zhu, Yuanhao Chen, Chenxi Lin, and Alan L. Yuille. “Max Margin Learning of Hierarchical Configurational Deformable Templates (HCDTs) for Efficient Object Parsing and Pose Estimation.” *IJCV*, **93**(1):1–21, 2011.
- [ZCT10] Long Zhu, Yuanhao Chen, Antonio Torralba, William T. Freeman, and Alan L. Yuille. “Part and appearance sharing: Recursive Compositional Models for multi-view.” In *CVPR*, pp. 1919–1926, 2010.
- [ZDG14] Ning Zhang, Jeff Donahue, Ross B. Girshick, and Trevor Darrell. “Part-Based R-CNNs for Fine-Grained Category Detection.” In *ECCV*, pp. 834–849, 2014.
- [ZF14] Matthew D. Zeiler and Rob Fergus. “Visualizing and Understanding Convolutional Networks.” In *ECCV*, pp. 818–833, 2014.

- [ZJR15] 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 *International Conference on Computer Vision (ICCV)*, 2015.
- [ZUS15] Yukun Zhu, Raquel Urtasun, Ruslan Salakhutdinov, and Sanja Fidler. “segDeepM: Exploiting Segmentation and Context in Deep Neural Networks for Object Detection.” In *CVPR*, 2015.