



THE UNIVERSITY
of ADELAIDE

**Learning Structured Prediction Models in
Computer Vision**

by

Fayao Liu

A thesis submitted in fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering, Computer and Mathematical Sciences
School of Computer Science

November 2015

Declaration

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

I give consent to this copy of my thesis when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

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

Signed: _____

Date: _____

Acknowledgements

First and foremost, I would like to express my sincere gratitudes to my principle supervisor, Prof. Chunhua Shen. I would have never been able to finish this thesis without his guidance. During the course of my PhD study, he has always been an encouraging, inspiring and patient mentor, from whom I've learned not only methodologies, but the way to look at problems. His intelligence, creativity and keen perceptions in cutting-edge research topics have deeply impressed me. He has also set me a good example by his diligence and continuous efforts, as well as rigorous scientific attitudes. There are numerous other things I've learned from him, which will continue guiding me in my future career.

I would like to thank my co-supervisors, Prof. Anton van den Hengel and Prof. David Suter. They have showed generous patience and continuous support throughout my PhD candidature. As the director of the Australian Centre for Visual Technologies (ACVT), Anton has provided me a good platform as well as opportunities to communicate and collaborate with many talented researchers. He has also given me a lot of help on improving my language skills. I appreciate his generosity, encouragement and enlightenment.

I would like to thank Prof. Ian Reid, who showed me in person how to organize and write a research paper by telling a story. His innovative perspectives and insightful advices have helped me to improve my understandings and paper writing. It has always been pleasant and inspiring discussing with him.

I owe special thanks to Dr. Guosheng Lin, who taught me structured learning and many other things. He impressed me by his profound professional expertise and his extraordinary persistence. I'm deeply grateful for his unselfish sharing and constant supporting throughout my PhD life.

My sincere thanks go to ACVT researchers, especially Dr. Qinfeng (Javen) Shi, Dr. Peng Wang and Dr. Sakrapee (Paul) Paisitkriangkrai, for their kindness and valuable suggestions. Talking and discussing with them have always benefited me a lot.

Many thanks to all my current and previous lab mates, with whom I spent the most important years of my life together. Especially, I would like to mention Quoc-Huy Tran, Zhen Zhang, Yongrui Qin, Lina Yao, Lei Luo, Chao Zhang. I will always treasure those days spent with them. I also owe thanks to my friends I do not list here, with whom I share excitements and frustrations.

Finally, special gratitudes are attributed to my family, to whom this thesis is dedicated to.

Publications

This thesis is based on the content of the following peer-reviewed conference and journal papers:

1. Fayao Liu, Chunhua Shen, Guosheng Lin; “*Deep Convolutional Neural Fields for Depth Estimation from a Single Images*”; In proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
2. Fayao Liu, Chunhua Shen, Guosheng Lin, Ian D. Reid; “*Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields*”; Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2015.
3. Fayao Liu, Guosheng Lin, Chunhua Shen; “*CRF Learning with CNN Features for Image Segmentation*”; Pattern Recognition (PR), 2015.
4. Fayao Liu, Guosheng Lin, Chunhua Shen; “*Structured Learning of Tree Potentials in CRF for Image Segmentation*”; Submitted to IEEE Transactions on Neural Networks and Learning Systems (TNNLS); Major Revision.

In addition, I have published or submitted the following papers:

1. Fayao Liu, Luping Zhou, Chunhua Shen, Jianping Yin; “*Multiple Kernel Learning in the Primal for Multimodal Alzheimer’s Disease Classification*”; In IEEE Journal of Biomedical and Health Informatics (JBHI), 2014.
2. Fayao Liu, Guosheng Lin, Chunhua Shen; “*Discriminative Training of Deep Fully-connected Continuous CRFs with Task-specific Loss*”; Submitted to IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
3. Fayao Liu, Ruizhi Qiao, Chunhua Shen, Lei Luo; “*From Kernel Machines to Ensemble Learning*”; Submitted to Pattern Recognition (PR).
4. Fayao Liu, Chunhua Shen, Ian Reid, Anton van den Hengel; “*Online Unsupervised Feature Learning for Visual Tracking*”; Submitted to Computer Vision and Image Understanding (CVIU).
5. Chunhua Shen, Junae Kim, Fayao Liu, Lei Wang, Anton van den Hengel; “*Efficient Dual Approach to Distance Metric Learning*”; In IEEE Transactions on Neural Networks and Learning Systems (TNNLS), 2014.

Abstract

Faculty of Engineering, Computer and Mathematical Sciences
School of Computer Science

Doctor of Philosophy

by Fayao Liu

Most of the real world applications can be formulated as structured learning problems, in which the output domain can be arbitrary, *e.g.*, a sequence or a graph. By modelling the structures (constraints and correlations) of the output variables, structured learning provides a more general learning scheme than simple binary classification or regression models. This thesis is dedicated to learning such structured prediction models, *i.e.*, conditional random fields (CRFs) and their applications in computer vision. CRFs are popular probabilistic graphical models, which model the conditional distribution of the output variables given the observations. They play an essential role in the computer vision community and have found wide applications in various vision tasks—semantic labelling, object detection, pose estimation, to name a few. Specifically, we here focus on two challenging tasks in this thesis: image segmentation (also referred as semantic labelling) and depth estimation from single monocular images, which represent two types of CRFs models—discrete and continuous. In summary, we made three contributions in this thesis.

First, we present a new approach to exploit tree potentials in CRFs for the task of image segmentation. This method combines the advantages of both CRFs and decision trees. Different from traditional methods, in which the potential functions of CRFs are defined as a *linear* combination of some pre-defined parametric models, we formulate the unary and the pairwise potentials as nonparametric forests—ensembles of decision trees, and learn the ensemble parameters and the trees in a unified optimization problem within the large-margin framework. In this fashion, we easily achieve *nonlinear* learning of potential functions on both unary and pairwise terms in CRFs. Moreover, we learn class-wise decision trees for each object that appears in the image. We further show that this challenging optimization can be efficiently solved by combining a modified column generation and cutting-planes techniques. Experimental results on both binary and multi-class segmentation datasets demonstrate the power of the learned nonlinear nonparametric potentials.

Second, we propose to model the unary potentials of the CRFs using a convolutional neural network (CNN). The deep CNN is trained on the large-scale ImageNet dataset and transferred to image segmentation here for constructing unary potentials of superpixels. The CRFs parameters are then learned within the max-margin framework using structured support vector machines (SSVM). To fully exploit context information in inference, we construct spatially related co-occurrence pairwise potentials and incorporate them into the energy function. This prefers labellings of object pairs that frequently co-occur in a certain spatial layout and at the same time avoids implausible labellings during the inference. Extensive experiments on binary and multi-class segmentation benchmarks demonstrate the potentials of the proposed method.

Third, different from the previous two works, we address the problem of continuous CRFs learning, applied to the task of depth estimation from single images. Specifically, we formulate and learn the unary and pairwise potentials of a continuous CRFs model with CNN networks in a unified framework. We term this new method as deep convolutional neural fields, abbreviated as DCNF. It jointly explores the capacity of deep CNN and continuous CRFs. The proposed method can be used for depth estimation of general scenes with no geometric priors nor any extra information injected. Specifically, in our case, the integral of the partition function can be calculated in a closed form such that we can exactly solve the log-likelihood maximization. Moreover, solving the inference problem for predicting depths of a test image is highly efficient as closed-form solutions exist. We then further propose an equally effective model based on fully convolutional networks and a novel superpixel pooling method, which is ~ 10 times faster, to speedup the patch-wise convolutions in the deep model. With this more efficient model, we are able to design very deep networks to pursue further performance gain. Experiments on both indoor and outdoor scene datasets demonstrate that the proposed method significantly outperforms state-of-the-art depth estimation approaches. We also show experimentally that the proposed method generalizes well to depth estimations of images unrelated to the training data. This indicates the potential of our method for benefiting other vision tasks.

Dedicated to my family.

Contents

Declaration	iii
Acknowledgements	v
Publications	vii
Abstract	ix
Contents	xiii
List of Figures	xvii
List of Tables	xxi
Notations	xxiii
1 Introduction	1
1.1 Structured Learning	2
1.2 Conditional Random Fields	2
1.2.1 Limitations of Current CRF Models	3
1.3 Contributions	4
2 Background Literature	7
2.1 Supervised Learning	7
2.1.1 Support Vector Machines	9
2.1.2 Logistic Regression	11
2.2 Structured Learning	12
2.2.1 Structured SVM	14
2.2.2 Conditional Random Fields	16
2.2.2.1 Continuous Conditional Random Fields	17
2.3 Ensemble Learning	17
2.3.1 Column Generation Boosting	18

2.4	Convolutional Neural Networks	20
2.4.1	CNN for Structured Predictions	21
3	CRF Learning with Tree Potentials for Image Segmentation	25
3.1	Introduction	26
3.1.1	Related Work	27
3.2	Segmentation Using CRF Models	29
3.3	Learning Tree Potentials in CRF	29
3.3.1	Energy Formulation	29
3.3.2	Learning CRF in the Max-Margin Framework	30
3.3.3	Learning Tree Potentials Using Column Generation	32
3.3.4	Speeding up Optimization Using Cutting-Plane	35
3.3.4.1	Implementation Details	36
3.3.4.2	Discussions on the Submodularity	37
3.4	Experiments	37
3.4.1	Experimental Setup	37
3.4.2	Comparing with Baseline Methods	38
3.4.2.1	Graz-02	38
3.4.2.2	MSRC-21	39
3.4.3	Comparing with State-of-the-art Methods	39
3.4.3.1	Weizmann Horse	40
3.4.3.2	Oxford Flower	40
3.4.3.3	Graz-02	41
3.4.3.4	MSRC-21	41
3.4.4	Object-aware vs. Non-object-aware	41
3.5	Conclusion	42
4	CRF Learning with CNN Potentials for Image Segmentation	49
4.1	Introduction	49
4.1.1	Related Work	51
4.2	Proposed Method	52
4.2.1	Deep Convolutional Neural Networks	52
4.2.2	Segmentation with CRF Models	53
4.2.3	Learning CRF in the Max-Margin Framework	54
4.2.3.1	Implementation Details	55
4.2.4	Inference with Co-Occurrence Pairwise Potentials	56
4.3	Experiments	57
4.3.1	Experimental Setup	57
4.3.2	Baseline Comparison	58
4.3.2.1	Weizmann Horse	59
4.3.2.2	Graz-02	59
4.3.2.3	MSRC-21	60
4.3.3	State-of-the-art Comparison	61
4.3.3.1	Binary Datasets	61
4.3.3.2	Multi-class Datasets	61
4.4	Conclusion	63

5	Joint Learning of Continuous CRF and CNN for Single Image Depth Estimation	71
5.1	Introduction	72
5.2	Related Work	74
5.2.1	Depth Perception in Vision	75
5.2.1.1	Depth Estimation from Single Monocular Images	75
5.2.2	Combining CNN and CRF	78
5.2.3	Fully Convolutional Networks	79
5.3	Deep Convolutional Neural Fields	80
5.3.1	Overview	80
5.3.2	Potential Functions	81
5.3.2.1	Unary potential	81
5.3.2.2	Pairwise Potential	82
5.3.3	Learning	83
5.3.3.1	Optimization	85
5.3.3.2	Depth Prediction	87
5.3.4	Speeding up Training Using Fully Convolutional Networks and Superpixel Pooling	88
5.3.4.1	DCNF-FCSP Overview	89
5.3.4.2	Fully Convolutional Networks	89
5.3.4.3	Superpixel Pooling	90
5.3.5	Implementation Details	91
5.4	Experiments	93
5.4.1	Baseline Comparisons	97
5.4.1.1	NYU v2 Dataset	98
5.4.1.2	Make3D Dataset	99
5.4.2	DCNF vs. DCNF-FCSP	100
5.4.3	State-of-the-art Comparisons	100
5.4.3.1	NYU v2 Dataset	100
5.4.3.2	Make3D Dataset	101
5.4.3.3	KITTI data	101
5.4.4	Generalization to Depth Estimations of General Scene Images	102
5.5	Conclusion	102
6	Conclusion	105
6.1	Future Work	106
6.1.1	Deep Structured Learning	106
6.1.2	Semi-supervised Structured Learning	107
	Bibliography	109

List of Figures

2.1	An illustration of the 0/1 loss upper bounded by the hinge loss and the log loss. The horizontal axis shows $\mathbf{w}^\top \Psi(\mathbf{x}, y) - \max_{y' \neq y} \mathbf{w}^\top \Psi(\mathbf{x}, y')$, where y is the correct label for the example \mathbf{x} , while the vertical axis quantifies the loss. As shown, the 0/1 loss is discontinuous, while the hinge loss is continuous; the log-loss is continuous and smooth. Figure reproduced from [1].	9
2.2	An illustration of the LeNet [2] for handwritten character recognition. Figure reproduced from [2].	20
2.3	(a) An illustration of a single convolutional layer followed by a pooling layer with pooling size being 2; (b) An illustration of a fully connected layer with 3 hidden neurons.	21
2.4	An illustration of the (a) low-level, (b) mid-level and (c) high-level features learned from different layers of CNN models. Figure reproduced from [3].	23
3.1	Segmentation examples produced by our model on images from the Oxford 17 Flower dataset with different column generation iterations. From left to right: Test images, 2nd, 4th, 6th, and 10th iteration.	26
3.2	Segmentation examples on the Weizmann horse dataset. 1st and 4th columns: Test images; 2nd and 5th columns: Ground truth; 3rd and 6th columns: Predictions produced by our CRFTree method.	43
3.3	Segmentation examples on MSRC. 1st column: Test images; 2nd column: Ground truth; 3rd column: Predictions of AdaBoost; 4th column: Predictions of SVM; 5th column: Predictions of SSVM; 6th column: Predictions of CRFTree with unsupervised feature learning.	44
3.4	Qualitative comparison on the Graz-02 dataset. 1st column: Test images; 2nd column: Ground truth; 3rd column: Predictions of AdaBoost; 4th column: Predictions of SVM; 5th column: Predictions of SSVM; 6th column: Predictions of CRFTree. SSVM and CRFTree present more smooth boundary than AdaBoost and SVM due to the introduce of pairwise terms. Compared to SSVM, our CRFTree yields more accurate segmentation because of the non-linearity property.	45
3.5	Examples of qualitative evaluations on the Oxford flower dataset. 1st and 4th columns: Test images; 2nd and 5th columns: Ground truth; 3rd and 6th columns: Predictions produced by our method CRFTree. Our predictions well preserve the boundaries.	46
3.6	Confusion matrices of the predictions of different models using bag-of-words feature and color histogram features on the MSRC dataset. (a) SSVM; (b) CRFTree.	47

4.1	An illustration of the proposed segmentation pipeline. We first over-segment the image into superpixels and then compute deep convolutional features of the patch around each superpixel centroid using a pre-trained deep CNN. The learned features are then used to learn a CRF for segmentation.	52
4.2	An illustration of the deep CNN architecture used for ImageNet classification by Krizhevsky <i>et al.</i> [4]. The first convolutional layer filters the input image with 96 kernels of size $11 \times 11 \times 3$ with a stride of 4 pixels; the second convolutional layer takes the output of the first layer as input and filters it with 256 kernels of size $5 \times 5 \times 96$; each of the 3rd and 4th layer has 384 kernels of size $3 \times 3 \times 256$ and $3 \times 3 \times 384$ respectively; the 5th convolutional layer has 256 kernels of size $3 \times 3 \times 384$; the fully connected layers have 4096 kernels each and the last soft-max layer has 1000 neurons. A max-pooling layer follows the first, second and fifth layer.	54
4.3	Segmentation examples on Weizmann horse. 1st column: Test images; 2nd column: Ground truth; 3rd column: Predictions produced by SSVM based CRF learning with bag-of-words feature; 4th column: Predictions produced by SSVM based CRF learning with unsupervised feature learning; 5th column: Predictions produced by SSVM based CRF learning with the 6th layer CNN features.	65
4.4	Segmentation examples on the Graz-02 dataset. 1st column: Test images; 2nd column: Ground truth; 3rd column: Segmentation results produced by SSVM based CRF learning with bag-of-words feature; 4th column: Segmentation results produced by SSVM based CRF learning with unsupervised feature learning; 5th column: Segmentation results produced by SSVM based CRF learning with the 6th layer CNN features.	66
4.5	Segmentation examples on the MSRC-21 dataset. 1st column: Test images; 2nd column: Ground truth; 3rd column: Predictions produced by SSVM based CRF learning with bag-of-words feature; 4th column: Predictions produced by SSVM based CRF learning with unsupervised feature learning; 5th column: Predictions results produced by our method with co-occurrence pairwise potentials.	67
4.6	Segmentation examples on the Stanford Background dataset. 1st and 4th columns: Test images; 2nd and 5th columns: Ground truth; 3rd and 6th columns: Predictions produced by our method with co-occurrence pairwise potentials.	68
4.7	Segmentation examples on the PASCAL VOC 2011 dataset. 1st and 4th columns: Test images; 2nd and 5th columns: Ground truth; 3rd and 6th columns: Predictions produced by our method with co-occurrence pairwise potentials.	68
4.8	Failure examples on the VOC 2011 dataset. 1st row: Test images; 2nd row: Ground truth; 3rd row: Segmentation results produced by our method with co-occurrence pairwise potentials.	69
4.9	Confusion matrix of the predictions produced by our method for a single run on the StanfordBackground dataset.	69
4.10	Occurrence frequencies of different categories in the training data of the StanfordBackground dataset.	69
4.11	Confusion matrix of the predictions made by our method on the MSRC dataset.	70

4.12	Confusion matrix of the predictions produced by our method on the Pascal VOC 2011 dataset.	70
5.1	Examples of depth estimation results using the proposed deep convolutional neural fields model. First row: NYU v2 dataset; second row: Make3D dataset. From left to right: input image, ground-truth, our prediction.	72
5.2	An illustration of the box model based methods for room layout estimation. Figure reproduced from [5].	75
5.3	An illustration of the block model based methods for outdoor 3D scene understanding. Left: examples of extracted blocks; Right: examples of super-pixel based density estimation. Figure reproduced from [6].	75
5.4	An illustration of the non-parametric methods for depth estimation. Figure reproduced from [7].	77
5.5	An illustration of the probabilistic model based methods for depth estimation. Left: input image; Right: superpixels overlaid with an MRF. Figure reproduced from [8].	77
5.6	An illustration of our DCNF model for depth estimation. The input image is first over-segmented into superpixels. In the unary part, for a superpixel p , we crop the image patch centred around its centroid, then resize and feed it to a CNN which is composed of 5 convolutional and 4 fully-connected layers (details refer to Fig. 5.7). In the pairwise part, for a pair of neighboring superpixels (p, q) , we consider K types of similarities, and feed them into a fully-connected layer. The outputs of unary part and the pairwise part are then fed to the CRF structured loss layer, which minimizes the negative log-likelihood. Predicting the depths of a new image \mathbf{x} is to maximize the conditional probability $\Pr(\mathbf{y} \mathbf{x})$, which has closed-form solutions (see Sec. 5.3.3 for details).	79
5.7	Detailed network architecture of the unary part in Fig. 5.6.	80
5.8	An overview of the unary part of the DCNF-FCSP model. For the unary part, the input image is fed into a fully-convolutional network to produce convolution maps (d is the number of filters of the last fully-convolutional layer). The obtained convolution maps, together with the superpixel segmentation over the original input image, are fed to a superpixel pooling layer. The outputs are $n \times 1$ d dimensional feature vectors for each of the n superpixels, which are then followed by 3 fully-connected layers to produce the unary output \mathbf{z} . The pairwise part are omitted here since we use the same network architecture as in the DCNF model (Fig. 5.6). The unary output \mathbf{z} and the pairwise output \mathbf{R} are used as input to the CRF loss layer, which minimizes the negative log-likelihood (See Sec. 5.3.4 for details).	85
5.9	The fully convolutional network architecture used in Fig. 5.8. The network takes input images of arbitrary size and output convolution maps.	85
5.10	An illustration of the superpixel pooling method, which mainly consists of convolution maps upsampling and superpixel pooling. The convolution maps are upsampled to the original image size by nearest neighbor interpolations, over which the superpixel masking is applied. Then average pooling is performed within each superpixel region, to produce the n convolution features. n is the number of superpixels in the image. d is the number of channels of the convolution maps.	88

5.11	Examples of qualitative comparisons on the NYUD2 dataset (Best viewed on screen). Color indicates depths (red is far, blue is close). Our method yields visually better predictions with sharper transitions, aligning to local details.	93
5.12	Comparison of the whole model training time (network forward + backward) in seconds (in <i>log</i> scale) for one image on the NYU v2 dataset with respect to different numbers of superpixels per image. The DCNF-FCSP model is orders of magnitude faster than the DCNF model.	94
5.13	Comparison of the network forward time of the whole model during depth prediction (in seconds) for one image on the NYU v2 dataset with respect to different numbers of superpixels per image. The DCNF-FCSP model is significantly faster than the DCNF model.	94
5.14	Examples of depth predictions on the Make3D dataset (Best viewed on screen). Depths are shown in log scale and in color (red is far, blue is close).	96
5.15	Examples of depth predictions on the KITTI dataset (Best viewed on screen). Depths are shown in log scale and in color (red is far, blue is close).	97
5.16	Examples of depth predictions on general indoor scene images obtained from the Internet (First row: test images; second row: our depth predictions. Best viewed on screen). Depths are shown in log scale and in color (red indicates far and blue indicates close).	97
5.17	An illustration of the absolute error maps and the pixel-wise error histograms of our predictions (Left: NYU v2; Right: Make3D). The absolute error maps are shown in meters, with the color bar shown in the last row. For the error histogram plot, the horizontal axis shows the prediction error in meters (quantized into 20 bins), and the vertical axis shows the percentage of pixels in each bin.	98
5.18	Examples of depth predictions on general outdoor scene images obtained from the Internet (First row: test images; second row: our depth predictions. Best viewed on screen). Depths are shown in log scale and in color (red indicates far and blue indicates close).	99

List of Tables

3.1	The average intersection-over-union score and average pixel accuracy comparison on the Graz-02 dataset. We include the foreground and background results in the brackets. Our method CRFTree with nonlinear and class-wise potentials learning performs better than all the baseline methods.	39
3.2	Performance of different methods on the Weizmann Horse dataset.	39
3.3	Performance of different methods on the Oxford Flower dataset. Our method CRFTree performs better than the compared methods.	40
3.4	The average intersection-over-union score and average pixel accuracy of CRFTree by incorporating unsupervised feature learning method. We include the foreground and background results in the brackets.	40
3.5	Comparing with state-of-the-art methods on the Graz-02 dataset. We report the F-score (%) for each class and the average over classes. Our method CRFTree outperforms all the compared methods with a large margin.	41
3.6	Segmentation results on the MSRC dataset. We report the pixel-wise accuracy for each category as well as the average per-category scores and the global pixel-wise accuracy. (1) The upper part presents the comparison with baseline methods, which all use bag-of-words and color histogram features. Our method CRFTree gains impressive improvements over SSVM while far better than simple linear models. (2) The lower part shows the results of our method incorporated with unsupervised feature learning (denoted as CRFTree (FL)) compared to state-of-the-art methods on this dataset.	42
3.7	Compared results of the object-aware (denoted as CRFTree (OA)) and the non-object-aware (denoted as CRFTree (NOA)) models on the MSRC dataset. Using object-aware potentials learning yields better results, which demonstrates the strength of the proposed method.	42
4.1	Performance of different methods on the Weizmann horse dataset. CNN features perform significantly better than the traditional BoW feature and the unsupervised feature learning method, with features of the 6th layer performing marginally better than other compared layers. SSVM based CRF learning performs far better than SVM.	59
4.2	Compared results of the average intersection-over-union score and average pixel accuracy on the Graz-02 dataset. We include the foreground and background results in the brackets. CNN features perform significantly better than the traditional BoW feature and the unsupervised feature learning, with features of the 6th layer performing the best among the compared layers in both SVM and SSVM. SSVM based CRF learning performs far better than SVM.	60

4.3	Segmentation results on the MSRC-21 dataset. We report the pixel-wise accuracy for each category as well as the average per-category scores and the global pixel-wise accuracy (%). Deep learning performs significantly better than the BoW feature and the unsupervised feature learning, with SSVM based CRF learning using features of the 7th layer of the deep CNN achieving the best results. SSVM based CRF learning performs far better than SVM.	60
4.4	State-of-the-art comparison of segmentation performance (%) on the Weizmann horse dataset.	61
4.5	State-of-the-art comparison of segmentation performance (%) on the Graz-02 (right) dataset.	61
4.6	State-of-the-art comparison of global and average per-category pixel accuracy on the MSRC-21 dataset.	63
4.7	State-of-the-art comparison of global and average per-category pixel accuracy on the Stanford Background dataset.	63
4.8	Results of per-category and mean intersection-over-union score (%) on the PASCAL VOC 2011 validation dataset. Best results are bold faced.	63
4.9	Comparison of the mean intersection-over-union score (%) on the PASCAL VOC 2011 validation dataset.	63
5.1	Baseline comparisons on the NYU v2 dataset. Our method with the whole network training performs the best.	92
5.2	Baseline comparisons on the Make3D dataset. Our method with the whole network training performs the best.	92
5.3	Performance comparisons of DCNF and DCNF-FCSP on the NYU v2 dataset. The two models show comparable performance.	92
5.4	Performance comparisons of DCNF and DCNF-FCSP on the Make3D dataset. The two models perform on par in general.	92
5.5	State-of-the-art comparisons on the NYU v2 dataset. Our method performs the best in most cases. Note that the results of Eigen <i>et al.</i> [9] are obtained by using extra training data (in the millions in total) while ours are obtained using the standard training set.	95
5.6	State-of-the-art comparisons on the Make3D dataset. Our method performs the best. Note that the C2 errors of the Discrete-continuous CRF [10] are reported with an ad-hoc post-processing step (train a classifier to label sky pixels and set the corresponding regions to the maximum depth).	95
5.7	State-of-the-art comparisons on the KITTI dataset. Our method achieves the best RMS error. Note that the results of Eigen <i>et al.</i> [9] are obtained by using extra training data (in the millions in total) while ours are obtained using 700 training images. The results of Saxena <i>et al.</i> [8] are reproduced from [9].	95

Notation

Symbol	Description
$\mathbf{1}$	Column vector with all elements being 1.
$\mathbf{0}$	Column vector with all elements being 0.
\mathbf{I}	Identity matrix.
\mathbb{R}	Domain of real numbers.
<i>i.i.d.</i>	Abbreviation of independent and identically distributed.
$\langle \cdot, \cdot \rangle$	Inner product operation.
\odot	Stacking two vectors.
\otimes	Kronecker tensor.
$\text{Tr}(\cdot)$	Trace of a matrix.
$\ \cdot\ _2$	L_2 norm.
Superscript \top	Transpose.
$\delta(\cdot)$	Indicator function which equals 1 if the input is true and 0 otherwise.
C	Trade-off parameter.
m	Number of examples.
ξ	Vector of slack variables.
\mathbf{w}	Vector of model parameters.
\mathbf{x}	Input observation.
\mathbf{y}	Structured output label.
y	Scalar output label.
\mathcal{X}	Input domain.
\mathcal{Y}	Output domain.
\mathcal{N}	Set of nodes.
\mathcal{S}	Set of edges.
\mathcal{W}	Working set.
\mathcal{H}	Domain of weak learners/decision trees.
$g : \mathcal{X} \rightarrow \mathcal{Y}$	Structured prediction function.
$f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$	Scoring function.
$l : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$	General loss function.
$\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$	Structured loss function.

E	Energy function.
U	Unary potential function.
V	Pairwise potential function.
Ψ	Feature mapping function.
$\Psi^{(1)}$	Unary feature mapping function.
$\Psi^{(2)}$	Pairwise feature mapping function.
\Pr	Probability function.
Z	Partition function.
sgn	Sign function.
\tilde{h}	A weak learner.
$\tilde{h}^{(1)}$	A unary decision tree.
$\tilde{h}^{(2)}$	A pairwise decision tree.
$\mathbf{H}^{(1)}$	A group of unary decision trees.
$\mathbf{H}^{(2)}$	A group of pairwise decision trees.
