

Joint Semantic Segmentation and Depth Estimation with Deep Convolutional Networks

Arsalan Mousavian

George Mason University
4400 University Dr, Fairfax, VA 22030
amousavi@gmu.edu

Hamed Pirsiavash

University of Maryland Baltimore County
1000 Hilltop Cir, Baltimore, MD 21250
hpirsiav@umbc.edu

Jana Košecká

George Mason University
4400 University Dr, Fairfax, VA 22030
kosecka@gmu.edu

Abstract

Multi-scale deep CNNs have been used successfully for problems mapping each pixel to a label, such as depth estimation and semantic segmentation. It has also been shown that such architectures are reusable and can be used for multiple tasks. These networks are typically trained independently for each task by varying the output layer(s) and training objective. In this work we present a new model for simultaneous depth estimation and semantic segmentation from a single RGB image. Our approach demonstrates the feasibility of training parts of the model for each task and then fine tuning the full, combined model on both tasks simultaneously using a single loss function. Furthermore we couple the deep CNN with fully connected CRF, which captures the contextual relationships and interactions between the semantic and depth cues improving the accuracy of the final results. The proposed model is trained and evaluated on NYUDepth V2 dataset [23] outperforming the state of the art methods on semantic segmentation and achieving comparable results on the task of depth estimation.

ers were often not sufficiently well localized for accurate pixel-level segmentation. To achieve more accurate localization property, the final layers have been combined with fully connected CRF's [4] yielding notable improvements in the segmentation accuracy. Independent efforts explored the use of CNNs for depth estimation from a single view [8]. Most recent work of [7] showed that common network architecture can be used for problems of semantic segmentation, depth estimation and surface normal estimation. The authors have shown that by changing the output layer and the loss function, the same network architecture can be trained effectively for different tasks achieving state of the art performance of different benchmark datasets. In contrast, we train the same network under multi task loss for semantic segmentation and depth estimation and our experiments show that multi-task learning boosts the performance.

We follow this line of work further and postulate the simultaneous availability of the depth estimates can further improve the final labeling. To support that we present a new approach and model for simultaneous depth estimation and semantic segmentation from a single RGB image, where the two tasks share the underlying feature representation. To further overcome the difficulties of deep CNNs to capture the context and respect the low-level segmentation cues as provided by edges and pixel values, we integrate CNN with a fully connected Conditional Random Field (CRF) model and learn its parameters jointly with the network weights. We train the model on NYUDepth V2 [23] and evaluate the final quality of both semantic segmentation with estimated depth, without depth and depth estimation alone. The proposed approach outperforms the state of the art semantic segmentation methods [7, 17, 11] and achieves comparable

1. Introduction

Deep convolutional networks (CNNs) attracted a lot of attention in the past few years and have shown significant progress in object categorization enabled by the availability of large scale labeled datasets [13]. For semantic segmentation problem, which requires learning a pixel-to-pixel mapping, several approaches have been proposed, for handling the loss of resolution and generation of a pixel level labelling [17, 2]. The initial CNN models for semantic segmentation showed that the response maps in final lay-

results on the task of depth estimation in [7].

2. Related work

In the past few years, convolutional neural networks have been applied to many high level problems in computer vision with great success. The initial categorization approaches focused on assigning a single label to an image [13], followed by application of the same classification strategy to windows or region proposals generated by independent segmentation process [9]. In addition to classification problems, these models marked also great success for a variety of regression problems, including pose estimation, stereo, localization and instance level segmentation, surface normal segmentation and depth estimation. The initial architectures obtained by concatenating multiple convolutional layers followed by pooling were suitable for image classification or regression problems, where single label of vector valued output was sought. The earlier layers before the fully connected layers were also found effective as feature maps used for variety of other traditional computer vision tasks [1]. For the problem of semantic segmentation CNN approaches typically generated features or label predictions at multiple scales [5] and used averaging and superpixels and for obtaining the final boundaries. In [18] CNNs were applied to superpixels, which were directly classified using feedforward multilayer network. Alternative strategy by [11] used CNN features computed over RGB-D region proposals generated by low-level segmentation methods. These methods although initially successful relied on the availability of independent segmentation methods to either refine the results or to generate object proposals.

One of the first approaches to tackle the semantic segmentation as a problem of learning a pixel-to-pixel mapping using CNNs was the work of [17]. Their authors proposed to apply 1x1 convolution label classifiers at feature maps from different layers and averaging the results. Another line of approaches to semantic segmentation adopted an auto-encoder style architecture [19] [2] comprised of convolutional and deconvolutional layers. The deconvolutional part consists of unpooling and deconvolution layers where each unpooling layer is connected to its corresponding pooling layer on the encoding side. The convolutional layers remain identical to the architectures of [13] and [24] and the deconvolutional layers are trained. Authors in [19] formulated the problem of semantic segmentation of the whole image as collage of individual object proposals, but also use the deconvolution part to delineate the object shape at higher resolution inside of the proposal window. The object proposal hypotheses are then combined by averaging or picking the maximum to produce the final output.

The lack of context or the capability of generating more accurate boundaries were some of the typical shortcomings of the above mentioned CNN based semantic segmenta-

tion architectures. In the pre-CNN approaches to semantic segmentation Conditional Random Fields (CRF) have been used effectively and provided strong means for integrating the local multi-class predictions with context and local information captured by pixels and edges [14]. To incorporate the benefits of CRF's for semantic segmentation the authors in Chen et al [4] proposed to combine deep CNNs responses of the last convolutional layer with the fully connected CRF. They used the hole method of [10] to make the VGG network [24] denser and resized the label probability map using bilinear interpolation. The resized semantic probability map was then used in place unary potentials for a fully connected CRF proposed by [12]. In spite of exhibiting significant improvement over initial results in [17], the method of [4] trained the CNN part and fully connected CRF part independently. Some of the subsequent efforts following this improvement led to joint training of CNNs and CRFs. Zheng et al. [28] addressed this issue by transforming the mean field approximation of [12] to a sequence of differentiable operations which can be incorporated in the CNN training. They learned with back-propagation the compatibility term of two labels regardless of the cell location. In the follow up work of [16] authors addressed this shortcoming by learning the compatibility between pairs of labels while considering their relative spatial location.

Previously reviewed methods for semantic segmentation have been applied to either images or RGB-D images, demonstrating improvements when the depth channel was available [11, 20]. Separate line of work focused on single image depth estimation. Early works exploited constraints of structured man-made, mostly indoors environments and rich features [26, 22]. Saxena et al [21] considered general outdoor scenes and formulated the depth estimation as Markov Random Field (MRF) labeling problem, where depth was estimated using a large set of handcrafted features computed at multiple scales and hierarchical MRF. Attempts to revisit these problems using deep CNNs were considered by Eigen et al [8], where depth was estimated using two networks, which handled coarse and fine scale depth estimation. The input to the first network is the whole image and output is a coarse depth map, while the second network, takes the coarse depth map produced by the previous stage and an image patch at 1/4 input image scale to produce the fine details of the depth map. Liu et al [15] addressed the depth estimation problem as estimating a single floating-point number for each superpixel representing the depth of superpixel center. There are few works where both the semantic and depth cues jointly contribute to semantic understanding and scene layout. Zhang et al [27], segmented the car instances in an image and provided the depth ordering of each car instance. Closest to our work in trying to use both depth and semantic cues are [14] and [25]. The authors of [14] propose to estimate depth and semantic

category using an unbiased semantic depth classifier, whose output on a bounding box remains the same when the image and bounding box scales by α . In [25], a coarse depth map is estimated by a CNN and they add finer depth details by extracting frequent templates for each semantic category. Finding frequent discriminant patches for each category requires more number of images from each category. As a result, their method does not scale well with the increase in number of classes.

The proposed model is the first to estimate the semantic labels and depth jointly from a single RGB image using a shared representation. While previous methods coupled CNNs with CRFs and refined the parameters of both components jointly, our approach is the first to do so with a more expressive objective function which incorporates the interactions between the depth and semantic labels.

3. Proposed Method

Semantic segmentation and depth estimation have been often addressed separately in the past. In this work, we demonstrate the possibility of training a network for depth estimation and semantic segmentation together, where the two tasks learn a shared underlying feature representation. This has a number of benefits: First of all, a single network handles both tasks which results in reducing the amount of computation and memory footprint by sharing the parameters. Moreover, the performance of semantic segmentation is increased because the network implicitly learns underlying physics by estimating depth for each pixel.

The proposed method takes RGB image as an input and uses a single network to make an initial estimate of depth and the semantic label for each pixel. These estimates are then combined to produce a final semantic segmentation. Using the estimated depth helps to resolve confusions between similar semantic categories such as pillow vs sofa, book vs bookshelf, and so on. The parameters of multi-scale network is learned by optimizing a joint objective function for semantic segmentation and depth estimation. The learned weights can be used for both tasks individually or for both, jointly. The proposed approach is an alternative to methods which use the depth channel of RGB-D sensor as an input to the network [17]. The raw depth channel often provides missing or inaccurate values, which are replaced by the output of the in-painting algorithms [3]. On the other hand, estimated depth from the network does not have any missing values.

The proposed model is outlined in Fig 1. Our initial goal in training is characterized by optimizing the loss function defined jointly for semantic categories and depth estimates:

$$L = \lambda \times L_{sem} + L_{depth} \quad (1)$$

In the above loss formulation L_{depth} and L_{sem} are optimized jointly using a shared representation in a multi-scale

CNN model, yielding a per pixel response maps of predicted labels and depth estimates. In the final stage of optimization the interactions between these response maps will be incorporated in a joint CRF model and the whole model including the network parameters will be further refined to minimize the objective. The following two sections will introduce the network and described the details of the individual loss functions L_{sem} and L_{depth} and how they related to the network structure. In Section 3.4 we will elaborate on the CRF formulation.

3.1. The model

The network has two main modules; one for semantic segmentation and one for depth estimation. Both modules use the same set of features to accomplish their task. The shared part of a network, which is shown in blue in Fig 1, is a multi-scale network that extracts features from images. It has been shown in the past that multi-scale networks are effective in improving the performance of semantic segmentation, which is analogous to extraction of features at multiple scales [4][7] in the traditional semantic segmentation approaches. The convolutional feature maps in the last layers of each scale are shared between semantic segmentation and depth estimation branches which are shown in green and red in Fig 1 respectively. The computed feature maps at different scales are upsampled and concatenated to form the comprehensive feature representation of the image. We chose to use the architecture of [4] because it produces denser output with stride of 8 using the atrous algorithm and has smaller memory footprint. Feature sharing results in saving computational resources during test time and also boosts the performance as shown in Section 4.

3.2. Semantic Loss

For semantic segmentation module the network outputs a response map with the dimension of $C \times H \times W$ where C is the number of semantic classes and H, W are the height and width of input image. The semantic segmentation loss is accumulated per-pixel multinomial logistic loss which is equal to

$$L_{sem} = - \sum_{i=1}^N \log(p(C_i^* | z_i)) \quad (2)$$

where C_i^* is the ground truth label of pixel i , $p(C_i^* | z_i) = e^{z_i} / \sum_c e^{z_{i,c}}$ is the probability of estimating semantic category C_i at pixel i , and $z_{i,c}$ is the output of the response map.

3.3. Depth Loss

In order to estimate the depth value, we divide the range of possible depth values to N_d bins where each bin has length l . For each bin b , the network predicts $p(b|x(i)) =$

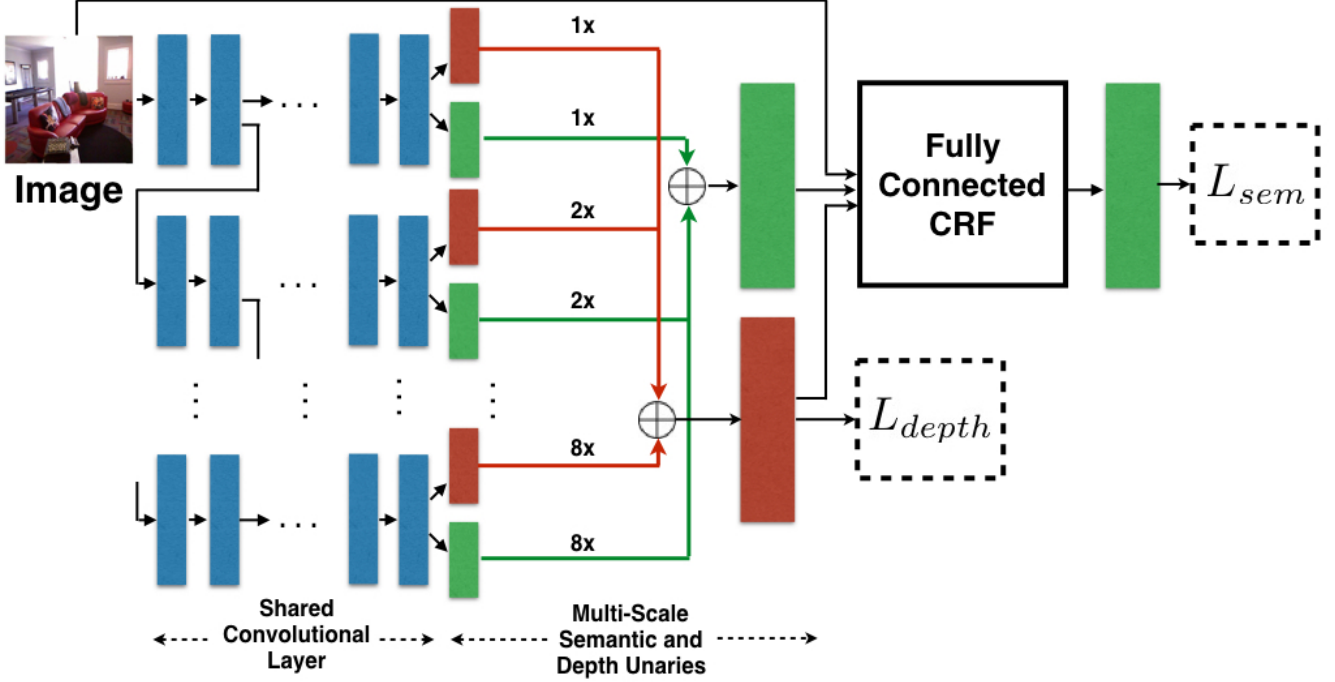


Figure 1. Overview of the proposed method. Multi-scale fully convolutional network is used for image representation. The network consists of 5 different paths and each path extracts feature at a difference scale. At the end of each path, two convolutional layers extract feature for semantic segmentation and depth estimation. These feature maps are scaled and aggregated to form a comprehensive feature map for semantic segmentation and depth estimation. Depth values are estimated using Eq. 3. Estimated depth values along with semantic unaries and the image are passed through the fully connected CRF to get the final probabilities of semantic labels.

$e^{r_i} / \sum_b e^{r_{i,b}}$, the probability of having an object at the center of that bin and $r_{i,b}$ is the response of network at pixel i and bin b . The continuous depth value d_i is the computed as:

$$d_i = \sum_{b=1}^{N_d} b \times l \times p(b|x(i)). \quad (3)$$

One might think that it should be also possible to learn the discretized depth probability using multinomial logistic loss similar to semantic segmentation. In this case however the training diverges due to following reasons; (1) multinomial softmax loss is not suitable for depth because depth is a continuous quantity and it cannot properly account for the distance of the estimated depth to the ground truth (it just indicates the estimated depth is incorrect); (2) estimating absolute depth for each pixel is ambiguous due to absence

of scene scale. Therefore we use scale-invariant loss function of [8] for depth estimation that tries to equalize the relative depth distance between any pair of points in the ground truth and the estimated depth values. Scale-invariant loss is computed as follows:

$$L_{depth} = \frac{1}{n^2} \sum_{i,j} ((\log(d_i) - \log(d_j)) - (\log(d_i^*) - \log(d_j^*)))^2 \quad (4)$$

The advantage of scale invariant loss is that it encourages to predict the correct relative depth of the objects with respect to each other rather than absolute depth values. Since we are exploiting depth discontinuities in the CRF, scale invariant loss is suitable for our setup.

3.4. Conditional Random Field

As observed previously unary CNN based semantic segmentation results showed that the response maps/labels are often not sufficiently well localized to achieve pixel accurate segmentation. This and the capability of capturing more general contextual relationships between semantic classes led to initial proposals for incorporating CRF's. Using these observations, we integrate the depth and semantic label predictions in the CRF framework. The unary potentials are computed from semantic output of the multi-scale network and pairwise terms are Gaussian potentials based on depth discontinuities, difference in RGB values of pixels and the compatibility between semantic labels. Let N be the number of pixels and $X = \{x_1, x_2, \dots, x_N\}$ be the label assignment and $x_i \in \{1, \dots, C\}$. The features that we are using for each pixel i are denoted by $f_i = \{p_i, I_i, d_i\}$ where p_i is the spatial location of pixel i , I_i is the RGB value of pixel i , and d_i is the estimated depth at pixel i . The energy function for the fully connected CRF is follows:

$$E(x, f) = \sum_i \psi_u(x_i) + \sum_{i,j} \psi_p(x_i, f_i, x_j, f_j) \quad (5)$$

where unary potentials $\psi_u(x_i)$ come from the multi-scale network (the left big green rectangle in Fig 1) and the pairwise potentials have the form

$$\psi_p(x_i, f_i, x_j, f_j) = \mu(x_i, x_j)k(f_i, f_j) \quad (6)$$

where $\mu(x_i, x_j)$ represents the compatibility function between semantic label assignments of pixel i and j . Gaussian kernel $k(f_i, f_j)$ adjusts the evidence that should be propagated between x_i and x_j based on the spatial distance, RGB distance, and depth distance between pairs of pixels. $k(f_i, f_j)$ consists of three different weights $\{w^{(i)} | i \in \{1, 2, 3\}\}$ where each w^i has $C \times C$ parameters that are being learned for all the pairs of semantic categories. Gaussian kernels also have hyper-parameters $\theta_{(\cdot)}$ that control the tolerance with respect to difference in depth values, RGB pixel values and spatial location of pairs of pixels. $k(f_i, f_j)$ is computed using the following equation:

$$\begin{aligned} k(f_i, f_j) = & w^{(1)} \exp \left(\frac{|p_i - p_j|^2}{2\theta_\alpha^2} + \frac{|I_i - I_j|^2}{2\theta_\beta^2} \right) \\ & + w^{(2)} \exp \left(\frac{|p_i - p_j|^2}{2\theta_\gamma^2} + \frac{|d_i - d_j|^2}{2\theta_\zeta^2} \right) \\ & + w^{(3)} \exp \left(\frac{|p_i - p_j|^2}{2\theta_\tau^2} \right) \end{aligned} \quad (7)$$

The inference in the CRF is done using mean-field approximation similar to [28]. In the CRF training stage both the compatibility terms, the kernel weights and unary potentials are learned in a single optimization procedure. The

derivatives are back propagated through the network further refining the shared feature representation captured by network weights. Note that the CRF only adjusts its weights and back-propagates the error only to the semantic unaries and shared layers through the semantic module. Estimated depths are only taken as extra input modality in the CRF. However since both L_{sem} and L_{depth} is still being optimized the depth convolution layers will be adjusted to keep the output depth values valid. In the following section, we present additional details of multi-stage optimization and the scrutinize the effects of different components of the loss function on the overall performance. Qualitative results are shown in Fig. 2 and Fig. 3. Fig 2 shows the qualitative results of joint depth estimation and semantic segmentation. It is worth noting that there are some cases where our network detects a category correctly but that category is labeled incorrectly in the dataset. Two examples of such situations are the the left window and the leftmost chair in front of the desk in the second and third rows of Fig. 2. Fig 3 shows qualitative effect of the CRF module on the output of semantic segmentation.

4. Experiments

Before we proceed with details on the performance evaluation, we present in more detail the parameters of the network. The shared part of a network, which is shown in blue in Fig. 1 is a multi-scale network that extracts features from the images. The details about the parameters of the layers are found in Table 1. The first dimension is the number of channels for the output and the rest is the kernel size of that layer. The network has 5 different branches each either takes an image or one of the earlier layers as input and computes more higher-level features. The input resolution is 513×513 and at the end of each branch the computed features for semantics and depth are resized so to the dimension of the image size.

4.1. Training Details

Training is done at multiple stages. The training objective function for stage 1 is only L_{sem} and for the rest of the stages, Eq. 1 is optimized for the training. In the first stage of training, the network is trained for 160K iterations with learning rate of $1e-10$, weight decay of 0.0005 and momentum of 0.99 for semantic segmentation. The network weights of stage 1 are initialized from the model of [4] which is pre-trained on MS-COCO dataset and fine-tuned on Pascal-VOC dataset.

In the second stage, the depth layers (shown in red in Fig 1) are added to the network that is already trained on semantic segmentation. The network is initialized with the previous stage weights and is trained using combined semantic segmentation and depth estimation loss for 10K iterations. The scale of semantic and depth loss are different. There-

Table 1. Details of multi-scale network for computing depth and semantic unaries. Dimensions of each layer shown in the number of output channels and the kernel size.

Branch	Input								
Branch1	RGB	conv1-1 64x3x3	conv1-2 64x3x3	conv1-seg 40x1x1	conv1-depth 50x1x1				
Branch2	RGB	conv2-1 64x3x3	conv2-2 64x3x3	pool2 64x3x3	conv2-3 128x3x3	conv2-seg 40x1x1	conv2-depth 50x1x1	upsample x2	
Branch3	pool2	conv3-1 128x3x3	conv3-2 128x3x3	pool3 128x3x3	conv3-3 128x3x3	conv3-4 128x3x3	conv3-seg 40x1x1	conv3-depth 50x1x1	upsample x4
Branch4	pool3	conv4-1 256x3x3	conv4-2 256x3x3	pool4 256x3x3	conv4-3 128x3x3	conv4-4 128x3x3	conv4-seg 40x1x1	conv4-depth 50x1x1	upsample x4
Branch5	pool4	conv5-1 512x3x3	conv5-2 512x3x3	pool5 512x3x3	conv5-3 1024x3x3	conv5-4 1024x1x1	conv5-seg 40x1x1	conv5-depth 50x1x1	upsample x8

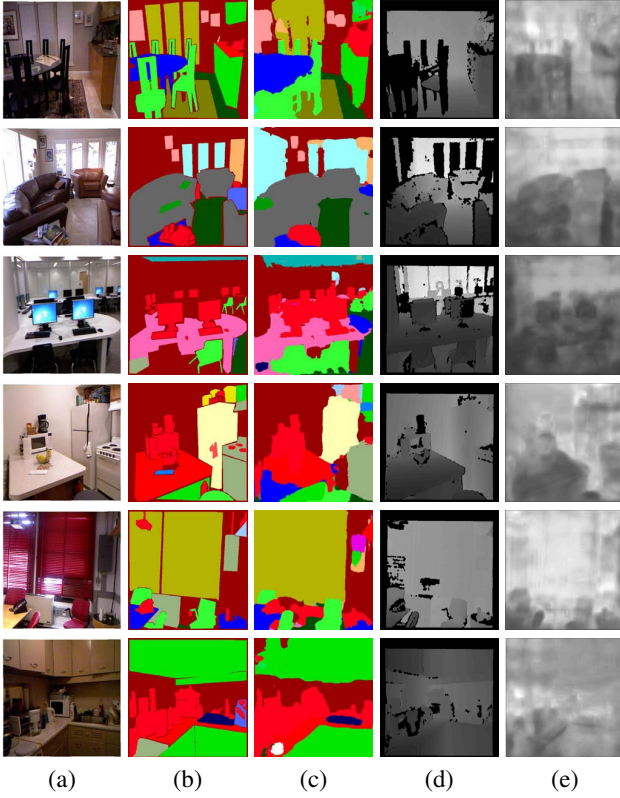


Figure 2. Qualitative result of the proposed method. (a) is the input image (b) is the ground truth semantic segmentation (c) is the output of our semantic segmentation (d) is the raw depth and (e) is the estimated depths. Note that in the second rows our network detects the left window correctly whereas it is labeled as wall in the ground truth. The same situation happens in the third row where the left black chair is missing in the ground truth but our network detects it. The dark black region in the ground truth depth are the missing depths. However, we do not have to deal with missing depths in our output.

fore, the effect of these loss functions should be balanced through the weight λ in Eq. 1. The λ was set to $1e-6$ to balance semantic loss and depth loss objectives. We also tried training with L_{depth} and L_{sem} together instead of two

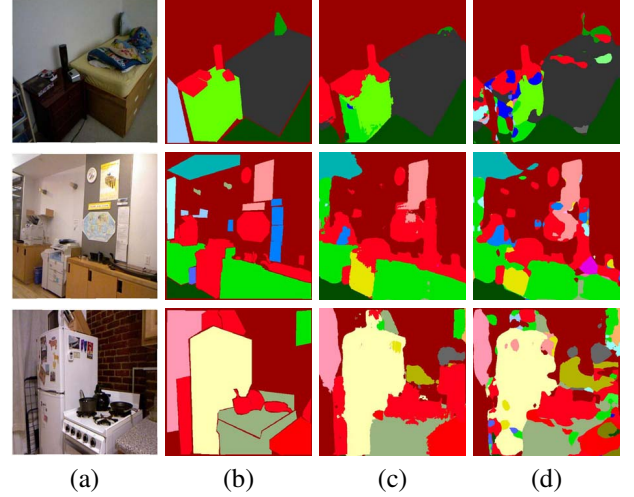


Figure 3. Qualitative comparison of with and without CRF on semantic segmentation. (a) is input image, (b) is ground truth labeling, (c) is semantic segmentation with CRF, and (d) is the semantic unaries without CRF.

stages of training. We observed that with the joint training, the value of objective function dropped much quicker but plateaued at the end. The two-stage training resulted in a slightly better model.

In the third stage, the fully connected CRF was added to the network fine-tuning the network jointly to learn the CRF weights. We used learning rate of $1e-13$ for the CRF weights and learning rate of $1e-16$ for the rest of network and ran the training for 10K iterations. In order to train the CRF, $w^{(1)}$ is initialized to 7, $w^{(2)}$ to 4, and $w^{(3)}$ is initialized with 3. The remaining parameters θ_α to 160, θ_β to 3, θ_γ to 50, θ_ζ to 0.2, and θ_τ to 3. All the initialization and hyper-parameters are found by cross validation on a random subset of 100 images from training set.

We trained and evaluated the model on NYUDepth v2 dataset [23] using the standard train/test split. The training set contains 795 images and the test set contains 654 images. For training the dataset is augmented by cropping, and mirroring. For each image, we generated 4 different crops and scale the depth accordingly. In addition, the orig-

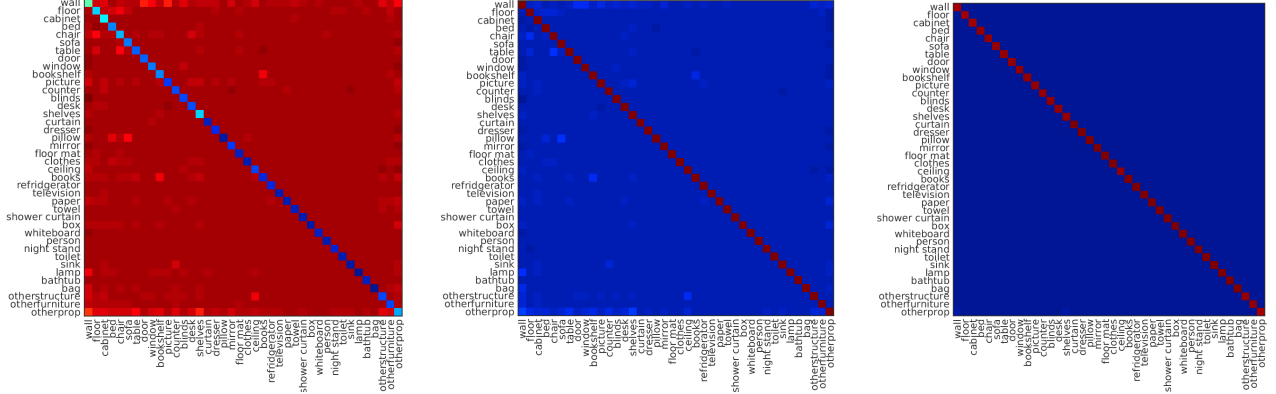


Figure 4. Visualization of learned weights in CRF. Left: compatibility function $\mu(.,.)$ between classes, middle: learned weights $w^{(2)}$ for depth for all pairs of semantic classes, right: learned weights $w^{(1)}$ for difference in RGB value of each pixel for all pairs of semantic classes (best viewed electronically).

inal image and its mirrored image were also included in the training set, yielding 4770 images from original training set. The data augmentation procedure was done offline and the data was shuffled randomly once before the training. The following sections contains the evaluation of our method on depth estimation and semantic segmentation.

4.2. Depth Estimation

For depth estimation, we use $N_d = 50$ bins with the length of $l = 0.14\text{m}$ in the network. After applying softmax and using Eq 3, depth value is estimated. We modified the ground truth depth values in the training to make the joint problem of semantic segmentation and depth estimation less complex and also reduce the uncertainty from the depth readings. Note that the ground truth depth values of evaluation set remain intact. The ground truth depth values are clipped at 7m because the quality of raw depth values from RGB-D decreases with the depth and the farther sensor readings are not reliable. We also rounded the depth value to the closest multiplier of l . We only used the valid depth values for training. Quantitative evaluation of our method is shown in Table 2. Our method outperforms only on the scale invariant loss which is sensitive to relative order of the entities with respect to each other. Given that the network is trained under multiple objective functions and learning relative ordering of the object is enough for reasoning in semantic space, it is reasonable that the network performs well only on scale invariant loss.

1. Percentage of Depth d_i where the ratio of estimated and ground truth depth is less than a threshold. i.e. $\max(\frac{d_i}{d_i^*}, \frac{d_i^*}{d_i}) = \delta < \text{threshold}$.
2. Absolute Relative Difference: $\frac{1}{T} \sum |d_i - d_i^*|/d_i^*$
3. Squared Relative Difference: $\frac{1}{T} \sum |d_i - d_i^*|^2/d_i^*$

4. RMSE (linear): $\sqrt{\frac{1}{|T|} \sum ||d_i - d_i^*||^2}$
5. RMSE (log): $\sqrt{\frac{1}{|T|} \sum ||\log(d_i) - \log(d_i^*)||^2}$
6. RMSE (log scale-invariant): equals to RMSE (log) after equalizing the mean estimated depth and ground truth depth.

where d and d^* are the estimated depth and ground truth depth respectively. Note that our RMSE error for scale invariant is significantly better and it quantitatively shows that our method is much better in finding depth discontinuities because scale invariant error, as the name implies, emphasizes on the relative depth not the absolute value of depth.

4.3. Semantic Segmentation

Semantic segmentation was evaluated on 40 semantic labels of NYUDepth V2 dataset using the mean Intersection over Union (IoU) which is the average Jaccard score among all the classes. Mean accuracy is the average pixel accuracy among all classes and pixel accuracy is the total accuracy of pixels regardless of the category. As shown in Table 3, our method outperforms the recent methods. Our-Unary-Sem is the performance of the network when only trained on semantic segmentation without depth (Training Stage 1). Ours-Unary-Sem+Depth is the network with semantic and depth without depth (Training Stage 2). Ours-Sem-CRF is the result of having both semantic and depth unaries but the CRF uses only RGB pixel values and semantic unaries as input. Our-Sem-CRF+ is including all the modules and CRF takes both the estimated depth and RGB pixel values as input. Overall, estimating the depth in addition to semantic segmentation improves the mean IoU over 40 classes by 1.3%. Similar observation is reported in [25], however our method is 10x faster and everything is trained end-to-end.

In order to further investigate how the CRF uses the depth information, $w^{(1)}$ and $w^{(2)}$ are visualized in Fig 4.

Table 2. Quantitative Evaluation of Depth Estimation

	Eigen et al.[7]	Liu et al [15]	Ours	
threshold $\delta < 1.25$	0.769	0.614	0.568	higher
threshold $\delta < 1.25^2$	0.950	0.883	0.856	is
threshold $\delta < 1.25^3$	0.988	0.971	0.956	better
abs relative distance	0.158	0.230	0.200	lower is better
sqr relative distance	0.121	-	0.301	
RMSE (linear)	0.641	0.824	0.816	
RMSE (log)	0.214	-	0.314	
RMSE (log. scale invariant)	0.171	-	0.061	

Table 3. Quantitative Evaluation of Semantic Segmentation on 40 Categories of NYUDepth v2

Method	Input Type	Mean IoU	Mean Accuracy	Pixel Accuracy
Deng at all [6]	RGBD	N/A	31.5	63.8
FCN[17]	RGB	29.2	42.2	60.0
FCN + Depth [17]	RGBD	34.0	46.1	65.4
Eigen and Fergus [7]	RGB	34.1	45.1	65.6
Ours-Unary-Sem	RGB	36.0	49.1	66.0
Ours-Unary-Sem+Depth	RGB	36.5	49.2	66.6
Ours-Sem-CRF	RGB	38.4	51.2	68.0
Ours-Sem-CRF+	RGB	39.2	52.3	68.6

Note that the difference in RGB values is not informative as the weights for differences in depth values between pixels. One interesting observation is that $w^{(2)}$ is large for pairs of classes where the depth discontinuity helps. Some of the examples of such pairs are pillow vs couch, bookshelf vs book, and sink vs counter.

5. Conclusions

We showed how to do semantic segmentation and depth estimation jointly using the same network which is trained in stages and then fine tuned using a single loss function. The proposed model and the training procedure produces comparable depth estimates and superior semantic segmentation comparing to state-of-the-art methods. Moreover, we showed that coupling CRF with the deep network further improves the performance and enables us to exploit the estimated depth to discriminate between some of the semantic categories. Our results show that depth estimation and semantic segmentation can share the underlying feature representations and can help to improve the final performance.

6. Acknowledgements

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory contract FA8650-12-C-7212. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorse-

ments, either expressed or implied, of IARPA, AFRL, or the U.S. Government. We also acknowledge support from NSF NRI grant 1527208. Some of the experiments were run on ARGO, a research computing cluster provided by the Office of Research Computing at George Mason University.

References

- [1] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. *CoRR*, abs/1406.5774, 2014. **2**
- [2] V. Badrinarayanan, A. Handa, and R. Cipolla. SegNet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. In *arXiv preprint arXiv:1505.07293*, 2015. **1, 2**
- [3] C. Cadena and J. Kosecka. Semantic segmentation with heterogeneous sensor coverage. In *ICRA*, 2013. **3**
- [4] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*, 2015. **1, 2, 3, 5**
- [5] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. In *ICLR*, 2013. **2**
- [6] Z. Deng, S. Todorovic, and L. J. Latecki. Semantic segmentation of RGBD images with mutex constraints. In *ICCV*, 2015. **9**
- [7] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. **1, 2, 3, 8, 9**
- [8] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014. **1, 2, 4**

- [9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2
- [10] A. Giusti, D. Ciresan, J. Masci, L. Gambardella, and J. Schmidhuber. Fast image scanning with deep max-pooling convolutional neural networks. In *ICIP*, 2013. 2
- [11] S. Gupta, R. Girshick, P. Arbelaez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *ECCV*, 2014. 1, 2
- [12] P. Krahenbuhl and V. Koltun. Efficient inference in fully connected CRFs with gaussian edge potentials. In *NIPS*, 2011. 2
- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2
- [14] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *CVPR*, 2014. 2
- [15] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, 2015. 2, 8
- [16] Z. Liu, X. Li, P. Luo, C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015. 2
- [17] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 2, 3, 9
- [18] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feed-forward semantic segmentation with zoom-out features. In *CVPR*, 2015. 2
- [19] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015. 2
- [20] X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *CVPR*, 2012. 2
- [21] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3D scene structure from a single still image. volume 31, 2008. 2
- [22] A. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box in the box: Joint 3D layout and object reasoning from single images. In *ICCV*, 2013. 2
- [23] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, 2012. 1, 6
- [24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2
- [25] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille. Towards unified depth and semantic prediction from a single image. In *CVPR*, 2015. 2, 3, 7
- [26] X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *CVPR*, 2015. 2
- [27] Z. Zhang, A. G. Schwing, S. Fidler, and R. Urtasun. Monocular object instance segmentation and depth ordering with CNNs. In *ICCV*, 2015. 2
- [28] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 2, 5