PAD-Net: Multi-Tasks Guided Prediction-and-Distillation Network for Simultaneous Depth Estimation and Scene Parsing

Dan Xu[1], Wanli Ouyang[2], Xiaogang Wang[3], Nicu Sebe[1]
[1]The University of Trento, [2]The University of Sydney, [3]The Chinese University of Hong Kong
{dan.xu, niculae.sebe}@unitn.it wanli.ouyang@sydney.edu.au xgwang@ee.cuhk.edu.hk

Abstract

*Depth estimation and scene parsing are two particularly important tasks in visual scene understanding. In this paper we tackle the problem of simultaneous depth estimation and scene parsing in a joint CNN. The task can be typically treated as a deep multi-task learning problem [42]. Different from previous methods directly optimizing multiple tasks given the input training data, this paper proposes a novel multi-task guided prediction-and-distillation network (PAD-Net), which first predicts a set of intermediate auxiliary tasks ranging from low level to high level, and then the predictions from these intermediate auxiliary tasks are utilized as multi-modal input via our proposed multi-modal distillation modules for the final tasks. During the joint learning, the intermediate tasks not only act as supervision for learning more robust deep representations but also provide rich multi-modal information for improving the final tasks. Extensive experiments are conducted on two challenging datasets (*i.e. *NYUD-v2 and Cityscapes) for both the depth estimation and scene parsing tasks, demonstrating the effectiveness of the proposed approach.*

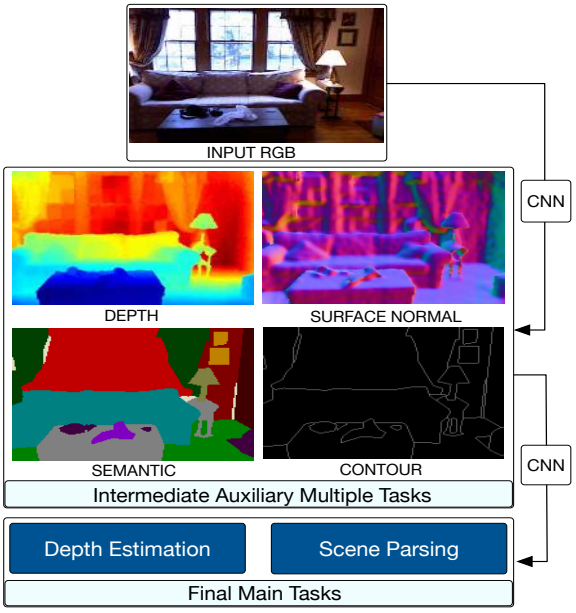

Figure 1. Motivation illustration. The proposed approach utilizes multiple intermediate multi-modal output from multi-task predictions as guidance to facilitate the final main-tasks. Different intermediate tasks ranging from low level to high level are considered, *i.e.* monocular depth prediction, surface normal estimation, contour prediction and semantic parsing.

1. Introduction

Depth estimation and scene parsing are both fundamental tasks for visual scene perception and understanding. Significant efforts have been made by many researchers on the two tasks in recent years. Due to the powerful deep learning technologies, the performance of the two individual tasks has been greatly improved [10, 54, 4]. Since these two tasks are correlated, jointly learning a single network for the two tasks is a promising research line.

Typical deep multi-task learning approaches mainly focused on the final prediction level via employing the cross-modal interactions to mutually refining the tasks [18, 51] or designing more effective joint-optimization objective functions [40, 21]. These methods directly learn to predict the two tasks given the same input training data. Under this setting, they usually require the deep models to partially share network parameters or hidden representations. However, simultaneously learning the different tasks using distinct loss functions makes the network optimization complicated, and it is generally not easy to obtain a good generalization ability for all the tasks, which therefore brings worse performance on some of the tasks compared with the optimization with only a single task, as found by UberNet [22]. In this paper, we explore multi-task deep learning from a different direction, *i.e.* using intermediate multi-task outputs as multi-modal input data. This is motivated by three observations. First, it is well-known that multi-modal data improve the performance of deep predictions. Take the task of scene parsing as an example, a CNN trained with

RGB-D data should perform better than the CNN trained with only the RGB data. If we do not have the depth data available, we can use a CNN to predict the depth maps and then use them as input. Second, instead of using the output only from the target tasks, *i.e*. semantic and depth maps, as the multi-modal input, the powerful CNN is able to predict more information related, such as contour and surface normal. Third, how to effectively use the multi-modal data obtained from intermediate auxiliary predictions to facilitates the final tasks is particularly important. In other words, it is a crucial point that how to design a good network architecture so that the network communicates or shares information based on the multi-modal data for different tasks, while other deep multi-task learning models such as Cross-stitch Net [38], Sluice Net [44], and Deep Relation Net [36], assume only single-modal data and thus do not consider it.

Based on the observations above, a multi-tasks guided prediction-and-distillation network (PAD-Net) is proposed. Specifically, we first learn to use a front-end deep CNN and the input RGB data to produce a set of intermediate auxiliary tasks (see Fig. 1). The auxiliary tasks range from low level to high level involving two continuous regression tasks (monocular depth prediction and surface normal estimation) and two discrete classification tasks (scene parsing and contour detection). The produced multiple predictions, *i.e*. depth maps, surface normal, semantic maps and object contours, are then utilized as the multi-modal input of the next sub-deep-network for the final two main tasks. By involving an intermediate multi-task prediction module, the proposed PAD-Net not only adds deep supervision for optimizing the front-end network more effectively, but also is able to incorporate more knowledge from relevant domains. Since the predicted multi-modal results are highly complementary, we further propose multi-modal distillation strategies to better using these data. When the optimization of the whole PAD-Net is finished, the inference is only based on the RGB input.

To summarize, the contribution of this paper is threefold: (i) First, we propose a new multi-tasks guided prediction-and-distillation network (PAD-Net) structure for simultaneous depth estimation and scene parsing. It produces a set of intermediate auxiliary tasks providing rich multi-modal data for learning the target tasks. Although PAD-Net takes only RGB data as input, it is able to incorporate multi-modal information for improving the final tasks. (ii) Second, we design and investigate three different multi-modal distillation modules for deep multi-modal data fusion, which we believe can be also applied in other scenarios such as multi-scale deep feature fusion. (iii) Third, extensive experiments on the challenging NYUD-v2 and Cityscapes datasets demonstrate the effectiveness of the proposed approach. Our approach achieves state-of-the-art results on NYUD-v2 on both the depth estimation and the scene parsing tasks, and obtains very competitive performance on the Cityscapes scene parsing task. More importantly, the proposed approach remarkably outperforms state-of-the-arts working on jointly optimizing both tasks.

2. Related Work

Depth estimation and scene parsing. The works on monocular depth estimation can be mainly grouped into two categories. The first group comprises the methods based on the hand-crafted features and graphical models [7, 45, 33]. For instance, Saxena *et al*. [45] proposed a discriminatively-trained Markov Random Field (MRF) model for multi-scale estimation. Liu *et al*. [33] built a discrete and continuous Conditional Random Field (CRF) model for fusing both local and global features. The second group of the methods is based on the advanced deep learning models [9, 32, 51, 43, 28]. Eigen *et al*. [10] developed a multi-scale CNN for fusing both coarse and fine predictions from different semantic layers of the CNN. Recently, researchers studied implementing the CRF models with CNN enabling the end-to-end optimization of the whole deep network [32, 54, 53].

Many efforts have been devoted to the scene parsing task in recent years. The scene parsing task is usually treated as a pixel-level prediction problem and the performance is greatly boosted by the fully convolutional strategy [35] which replaces the full connected layers with convolutional layers and dilated convolution [4, 58]. The other works mainly focused on multi-scale feature learning and ensembling [5, 52, 16], end-to-end structure prediction with CRF models [34, 1, 60, 55] and designing convolutional encoder-decoder network structures [41, 2]. These works focused on an individual task but not jointly optimizing the depth estimation and scene parsing together.

Some works [40, 51, 18, 26] explored simultaneously learning the depth estimation and the scene parsing tasks. For instance, Wang *et al*. [51] introduced an approach to model the two tasks within a hierarchical CRF, while the CRF model is not jointly learned with the CNN. However, these works directly learn the two tasks without treating them as multi-modal input for the final tasks.

Deep multi-task learning for vision. Deep multi-task learning[38, 44] has been widely used in various computer vision problems, such as joint inference scene geometric and semantic [21], face attribute estimation [14], simultaneous contour detection and semantic segmentation [12]. Yao and Urtasun *et al*. [57] proposed an approach for joint learning three tasks *i.e*. object detection, scene classification and semantic segmentation. Hariharan *et al*. [15] proposed to simultaneously learn object detection and semantic segmentation based on the R-CNN framework. However, none of them considered introducing multi-task prediction and multi-modal distillation steps at the intermediate level of a CNN to improve the target tasks.

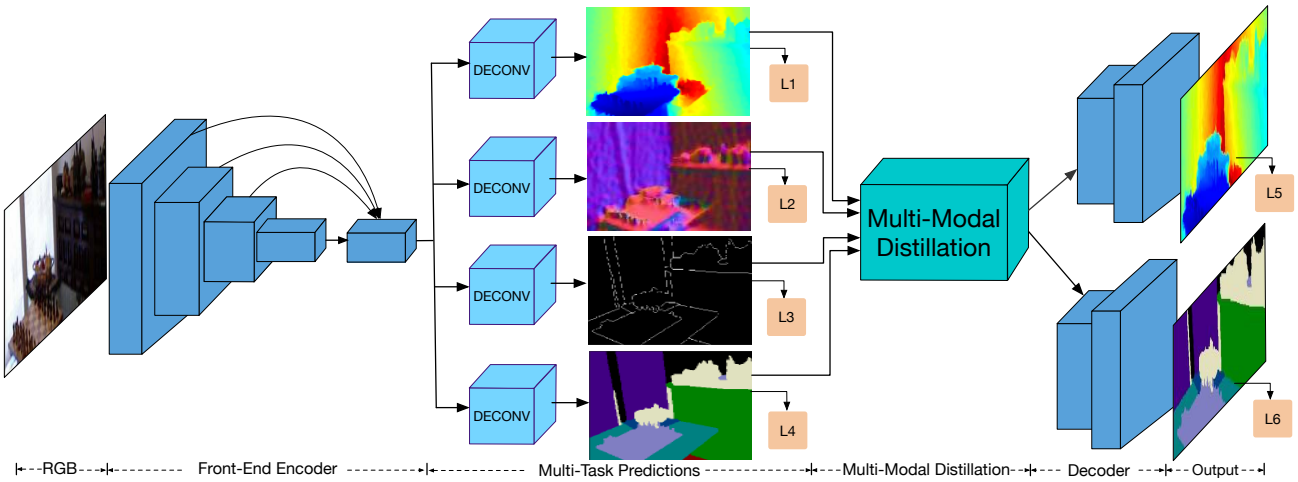

Figure 2. Illustration of the proposed PAD-Net for simultaneous depth estimation and scene parsing. The symbols of L1 to L6 denote different optimization losses for different tasks. 'DECONV' denotes the deconvolutional operation for upsampling and generating task-specific feature maps. The cube 'Multi-Modal Distillation' represents the proposed multi-modal distillation module for fusing the multiple predictions to improve the final main tasks.

3. PAD-Net: Multi-Tasks Guided Prediction-and-Distillation Network

In this section, we describe the proposed PAD-Net for simultaneous depth estimation and scene parsing. We first present an overview of the proposed PAD-Net, and then introduce the details of the PAD-Net. Finally, we illustrate the optimization and inference schemes for the overall network.

3.1. Approach Overview

Figure 2 depicts the framework of the proposed multi-tasks guided prediction and distillation network (PAD-Net). PAD-Net consists of four main components. First, a front-end fully convolutional encoder produces deep features. Second, an intermediate multi-task prediction module, which uses the deep features in the previous component for generating intermediate predictions. Third, a multi-modal distillation module which is used for incorporating useful multi-modal information from the intermediate predictions to improve the final tasks. Fourth, the decoders uses the distilled information for depth estimation and scene parsing. The input of PAD-Net is RGB images during both training and testing, and the final output is the depth and semantic parsing maps. During training, ground-truth labels for scene parsing, depth estimation and other two intermediate tasks, *i.e*. surface normal estimation and contour prediction, are used. Although four different kinds of supervision are used, we do not require extra annotation effort, since the surface normal and the contours can be directly inferred from depth and semantic labels, respectively.

3.2. Front-End Network Structure

The front-end backbone CNN could employ any network structures, such as the commonly used AlexNet [25], VGG [49] and ResNet [17]. To obtain better deep representations for predicting multiple intermediate tasks, we do not directly use the features from the last convolutional layer of the backbone CNN. A multi-scale feature aggregation procedure is performed to enhance the last-scale feature map via combining the previous scales feature maps derived from different semantic layers of the backbone CNN, as shown in Figure 2. The larger-resolution feature maps from shallower layers are down-sampled via convolution and bilinear interpolation operations to the resolution of the last-scale feature map. The convolution operations are also used to control the number of feature channels to make the feature aggregation more memory efficient. And then all the re-scaled feature maps are concatenated for the follow up deconvolutional operations. Similar to [3, 58], we also apply the dilated convolution strategy in the front-end network to produce feature maps with enlarged receptive field.

3.3. Deep Multi-Task Prediction

Using deep features from the front-end CNN, we perform deconvolutional operations to generate four sets of task-specific feature maps. We obtain features with N channels for the main depth estimation and scene parsing tasks while features with $N/2$ channels for the other two auxiliary tasks. The feature map resolution is made to be the same for four tasks and to be $2\times$ as that of the front-end feature maps. Then separate convolutional operations are performed to produce the score maps for the corresponding four tasks. The score maps are made to be $1/4$ as the resolution of the input RGB images via the bilinear interpolation. Four different loss functions are added for learning the four intermediate tasks with the re-scaled ground-truth maps. It should be noted that the intermediate multi-task learning not only provides deep supervision for optimizing the front-end CNN, but also helps to provide valuable multi-modal predictions, which are further used as input for the final tasks.

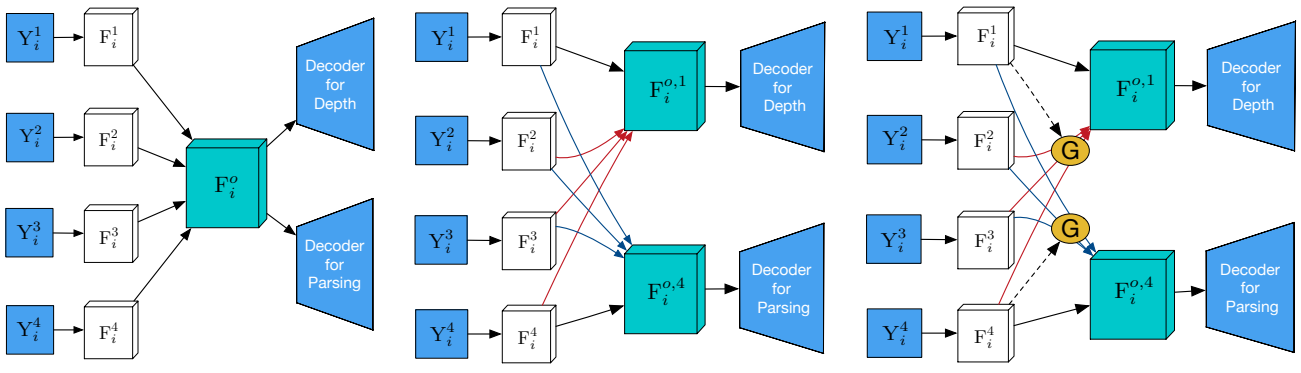

Figure 3. Illustration of the designed different multi-modal distillation modules. The symbols Y_i^1, Y_i^2, Y_i^3, Y_i^4 represent the predictions corresponding to multiple intermediate tasks. The distillation module A is a naive combination of the multiple predictions; the module B proposes a mechanism of passing message between different predictions; the module C shows an attention-guided message passing mechanism for distillation. The symbol G denotes a generated attention map which is used as guidance in the distillation.

3.4. Deep Multi-Modal Distillation

As mentioned before, the deep multi-modal distillation module fuses information from the intermediate predictions for each specific final task. It aims at effectively utilizing the complementary information from the intermediate predictions of relevant tasks. To achieve this target and under our general framework, it is potentially flexible to use any distillation scheme. In this paper, we develop and investigate three different module designs as shown in Figure 3 to show how the multi-modal distillation helps improving the final tasks. The distillation module A represents a naive concatenation of the features extracted from these predictions. The distillation module B passes message between different predictions. The distillation module C is an attention-guided message passing mechanism for information fusion. To generate richer information and bridge the gap between these predictions, before the distillation procedure, all the intermediate prediction maps associated with the i-th training sample, denoted by $\{\mathrm{Y}_i^t\}_{t=1}^T$, are first correspondingly transformed to feature maps $\{\mathrm{F}_i^t\}_{t=1}^T$ with more channels via convolutional layers, where T is the number of intermediate tasks.

Multi-Modal Distillation module A. A common way in deep networks for information fusion is to perform a naive concatenation of the feature maps or the score maps from different semantic layers of the network. We aslo consider this simple scheme as our basic distillation module. The module A outputs only one set of fused feature maps via $\mathrm{F}_i^o \leftarrow \mathrm{CONCAT}(\mathrm{F}_i^1, ..., \mathrm{F}_i^T)$, where $\mathrm{CONCAT}(\cdot)$ denotes the concatenation operation. And then F_i^o is fed into different decoders for predicting different final tasks, *i.e.* the depth estimation and the scene parsing tasks.

Multi-Modal Distillation module B. The module A outputs the same set of feature maps for the two final tasks. Differently, the module B learns a separate set of feature maps for each final task. For the k-th final task, let us denote F_i^k as the feature maps before message passing and denote $\mathrm{F}_i^{o,k}$ as the feature maps after the distillation. We refine F_i^k via passing message from the feature maps of other tasks as follows:

$$\mathrm{F}_i^{o,k} \leftarrow \mathrm{F}_i^k + \sum_{t=1(\neq k)}^{T} (\mathrm{W}_{t,k} \otimes \mathrm{F}_i^t), \tag{1}$$

where $\otimes$ denotes convolution operation, and $\mathrm{W}_{t,k}$ denotes the parameters of the convolution kernel corresponding to the t-th feature map and the k-th feature map. Then the obtained feature map $\mathrm{F}_i^{o,k}$ is used by the decoded for the corresponding k-th task. By using the task-specific distillation feature maps, the network can preserve more information for each individual task and is able to facilitate smooth convergence.

Multi-Modal Distillation module C. The module C introduces an attention mechanism for the distillation task. The attention mechanism [39] has been successfully applied in various tasks such as image caption generation [56] and machine translation [37] for selecting useful information. Specifically, we utilize the attention mechanism for guiding the message passing between the feature maps generated from different madalities for different tasks. Since the passed information flow is not always useful, the attention can act as a gate function to control the flow, in other words to make the network automatically learn to focus or to ignore information from other features. When we pass message to the k-th task, an attention map G_i^k is first produced from the corresponding set of feature maps F_i^k as follows:

$$\mathrm{G}_i^k \leftarrow \sigma(\mathrm{W}_g^k \otimes \mathrm{F}_i^k), \tag{2}$$

where W_g^k is the convolution parameter and σ is a sigmoid function for normalizing the attention map. Then the message is passed with the attention map controlled as follows:

$$\mathrm{F}_i^{o,k} \leftarrow \mathrm{F}_i^k + \sum_{t=1(\neq k)}^{T} \mathrm{G}_i^k \odot (\mathrm{W}_t \otimes \mathrm{F}_i^t), \tag{3}$$

where $\odot$ denotes element-wise multiplication.

3.5. Decoder Network Structure

For the task-specific decoders, we use two consecutive deconvolutional layers to up-sample the distilled feature maps for pixel-level prediction. Since the distilled feature maps have a resolution of $1/4$ to that of the input RGB image, each deconvolutional layer 2 time up-scales in resolution and accordingly reduces the number of output channels by half. Finally we use a convolution operation to generate the score maps for each final task.

3.6. PAD-Net Optimization

End-to-end network optimization. We have four intermediate prediction tasks, *i.e*. two discrete classification problems (scene parsing and contour prediction) and two continuous regression problems (surface normal estimation and depth estimation). However, we only require the annotations of the semantic labels and the depth, since the contour labels can be generated from the semantic labels and the surface normal can be calculated from the depth map. As our final target is to simultaneously perform the depth estimation and scene parsing, the whole network needs to optimize six losses with four different types. Specifically, we use a cross-entropy loss for the contour prediction task, a softmax loss for the scene parsing task and an Euclidean loss for both the depth and surface normal estimation tasks. Since the groundtruth depth maps have invalid points, we mask these points during training. Similar to previous works [47, 50], we jointly learn the whole network with a linearly combined optimization objective, *i.e*. $L_{all} = \sum_{i=1}^{6} w_i * L_i$, where L_i is the loss for the i-th task and w_i is the corresponding loss weight.

Inference. During the inference, We obtain the prediction results from the separate decoders. One important advantage of the PAD-Net is that it is able to incorporate rich domain knowledge from different predictions, *i.e*. scene semantic, depth, surface normal and object contours, while it only requires a single RGB image for the inference.

4. Experiments

To demonstrate the effectiveness of the proposed approach for simultaneous depth recovery and scene parsing, we conduct experiments on two publicly available benchmark datasets which provide both the depth and the semantic labels, including an indoor dataset NYU depth V2 (NYUD-v2) [48] and an outdoor dataset Cityscapes [6]. In the following we describe the details of our experimental evaluation.

4.1. Experimental Setup

Datasets and Data Augmentation. The **NYUD-v2** dataset [48] is a popular indoor RGBD dataset, which has been widely used for depth estimation [10] and semantic segmentation [13]. It contains 1449 pairs of RGB and depth images captured from a Kinect sensor, in which 795 pairs are used for training and the rest 654 for testing. Following [13], The training images are cropped to have a resolution of 560×425. The training data are augmented on the fly during the training phase. The RGB and depth images are scaled with a randomly selected ratio in $\{1, 1.2, 1.5\}$ and the depth values are divided by the ratio. We also flip the training samples with a possibility of 0.5.

The **Cityscapes** [6] is a large-scale dataset mainly used for semantic urban scene understanding. The dataset is collected over 50 different cities spanning several months, and overall 19 semantic classes are annotated. The fine-annotated part consists of training, validation and test sets containing 2975, 500, and 1525 images, respectively. The dataset also provides pre-computed disparity depth maps associated with the rgb images. Similar to NYUD-v2, we perform the data augmentation on the fly by scaling the images with a selected ratio in $\{0.5, 0.75, 1, 1.25, 1.75\}$ and randomly flipping them with a possibility of 0.5. As the images of the dataset have a high resolution (2048×1024), we crop the image with size of 640 for training due to the limitation of the GPU memory.

Evaluation Metrics. For evaluating the performance of the depth estimation, we use several quantitative metrics following previous works [10, 32, 54], including (a) mean relative error (rel): $\frac{1}{N}\sum_p \frac{|d_p - d_p^*|}{d_p}$; (b) root mean squared error (rms): $\sqrt{\frac{1}{N}\sum_p (d_p - d_p^*)^2}$; (c) mean log10 error (log10): $\frac{1}{N}\sum_i \|\log_{10}(d_p) - \log_{10}(d_p^*)\|$ and (d) accuracy with threshold t: percentage (%) of d_p^* subject to $\max(\frac{d_p^*}{d_p}, \frac{d_p}{d_p^*}) = \delta < t$ ($t \in [1.25, 1.25^2, 1.25^3]$), where d_p and d_p^* are the prediction and the groundtruth depth at the p-th pixel, respectively. For the evaluation of the semantic segmentation, we adopt three commonly used metrics, *i.e*. mean Intersection over Union (mIoU), mean accuracy and pixel accuracy. The mean IoU is calculated via averaging the Jaccard scores of all the predicted classes. The mean accuracy is the accuracy among all classes and pixel accuracy is the total accuracy of pixels regardless of the category. On the Cityscapes, both the pixel-level mIoU and instance-level mIoU are considered.

Implementation Details. The proposed network structure is implemented base on *Caffe* library [19] and on Nvidia Titan X GPUs. The front-end convolutional encoder of PAD-Net naturally supports any network structure. During the training, the front-end network is first initialized with parameters pre-trained with ImageNet for training, and the rest of the network is randomly intialized. The whole training process is performed with two phases. In the first phase, we only optimize the front-end network with the scene parsing task and use a learning rate 0.001. After that, the whole network is jointly trained with multi-task losses and a lower learning rate of 10e-5 is used for a smooth con-

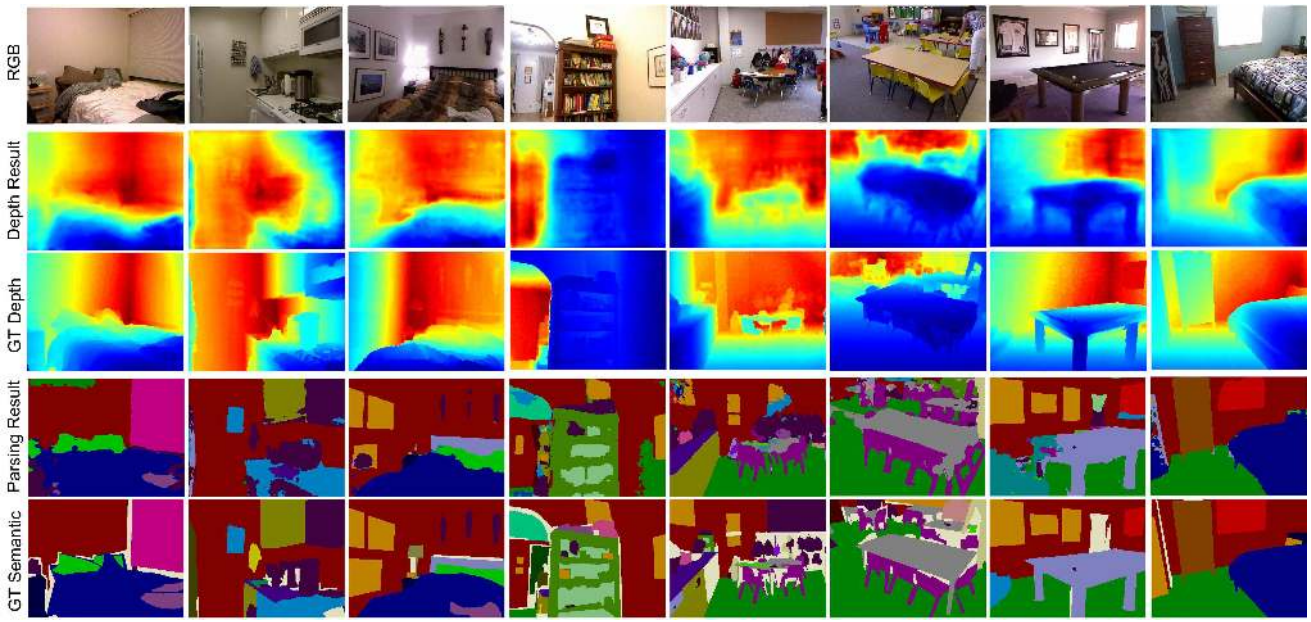

Figure 4. Quanlitative examples of depth prediction and 40-classes scene parsing results on the NYUD-v2 dataset. The second and the four row are the estimated depth maps and the scene parsing results from the proposed PAD-Net, respectively.

Table 1. Diagnostic experiments for the depth estimation task on NYUD-v2 dataset. Distillation A, B, C represents the proposed three multi-modal distillation modules.

Method	Error (lower is better)			Accuracy (higher is better)		
	rel	log10	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Front-end + DE (baseline)	0.265	0.120	0.945	0.447	0.745	0.897
Front-end + DE + SP (baseline)	0.260	0.117	0.930	0.467	0.760	0.905
PAD-Net (Distillation A + DE)	0.248	0.112	0.892	0.513	0.798	0.921
PAD-Net (Distillation B + DE)	0.230	0.099	0.850	0.591	0.854	0.953
PAD-Net (Distillation C + DE)	0.221	0.094	0.813	0.619	0.882	0.965
PAD-Net (Distillation C + DE + SP)	**0.214**	**0.091**	**0.792**	**0.643**	**0.902**	**0.977**

Table 2. Diagnostic experiments for the scene parsing task on the NYUD-v2 dataset.

Method	Mean IoU	Mean Accuracy	Pixel Accuracy
Front-end + SP (baseline)	0.291	0.301	0.612
Front-end + SP + DE (baseline)	0.294	0.312	0.615
PAD-Net (Distillation A + SP)	0.308	0.365	0.628
PAD-Net (Distillation B + SP)	0.317	0.411	0.638
PAD-Net (Distillation C + SP)	0.325	0.432	0.645
PAD-Net (Distillation C + DE + SP)	**0.331**	**0.448**	**0.647**

vergence. As the final tasks are depth estimation and scene parsing, we set the loss weight of the contour prediction and surface normal estimation as 0.8. In the multi-task prediction module, N is set to 512. Total 60 epochs are used for NYUD-v2, and 40 epochs for Cityscapes. Due to the sparse groundtruth depth maps of the Cityscapes dataset, the invalid points are masked out in the backpropagation. The network is optimized using stachastic gradient descent with the weight decay and the momentum set to 0.0005 and 0.99, respectively.

4.2. Diagnostics Experiments

To deeply analyze the proposed approach and demonstrate its effectiveness, we conduct diagnostics experiments

Table 3. Quantitative comparison with state-of-the-art methods methods on the scene parsing task on the NYUD-v2 dataset. The methods 'Gupta *et al.*' [13] and 'Arsalan *et al.*' [40] jointly learn two tasks.

Method	Input Data Type	Mean IoU	Mean Accuracy	Pixel Accuracy
Deng *et al.* [8]	RGB + Depth	-	0.315	0.638
FCN [35]	RGB	0.292	0.422	0.600
FCN-HHA [35]	RGB + Depth	0.340	0.461	0.654
Eigen and Fergus [9]	RGB	0.341	0.451	0.656
Context [31]	RGB	0.406	0.536	0.700
Kong *et al.* [23]	RGB	0.445	-	0.721
RefineNet-Res152 [30]	RGB	0.465	0.589	0.736
Gupta *et al.* [13]	RGB + Depth	0.286	-	0.603
Arsalan *et al.* [40]	RGB	0.392	0.523	0.686
PAD-Net-ResNet50 (Ours)	RGB	**0.502**	**0.623**	**0.752**

on both NYUD-v2 and Cityscapes datasets. For the front-end network, according to the complexity of the dataset, we choose AlexNet [25] and ResNet-50 [17] network structures for NYUD-v2 and Cityscapes, respectively.

Baseline methods and different variants of PAD-Net. To conduct the diagnostic experiments, we consider two baseline methods and different variants of the proposed PAD-Net. The baseline methods include: (i) Front-end + DE: performing the depth estimation (DE) task with the front-end CNN; (ii) Front-end + SP + DE: performing the scene parsing (SP) and the depth estimation tasks simultaneously with the front-end CNN. The different variants include: (i) PAD-Net (Distillation A + DE): PAD-Net performing the DE task using the distillation module A; (ii) PAD-Net (Distillation B + DE): similar to (i) while using the distillation module B; (iii) PAD-Net (Distillation B + DE): similar to (i) while using the distillation module C; (iv) PAD-Net (Distillation C + DE + SP): performing DE and SP tasks simultaneously with the distillation module C.

Table 4. Quantitative comparison with state-of-the-art methods on the depth estimation task on NYUD-v2 dataset. The methods 'Joint HCRF' [51] and 'Jafari *et al.*' [18] simultaneously learn the two tasks.

Method	# of Training	Error (lower is better)			Accuracy (higher is better)		
		rel	log10	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Saxena *et al.* [46]	795	0.349	-	1.214	0.447	0.745	0.897
Karsch *et al.* [20]	795	0.35	0.131	1.20	-	-	-
Liu *et al.* [33]	795	0.335	0.127	1.06	-	-	-
Ladicky *et al.* [27]	795	-	-	-	0.542	0.829	0.941
Zhuo *et al.* [61]	795	0.305	0.122	1.04	0.525	0.838	0.962
Liu *et al.* [32]	795	0.230	0.095	0.824	0.614	0.883	0.975
Eigen *et al.* [10]	120K	0.215	-	0.907	0.611	0.887	0.971
Roi *et al.* [43]	795	0.187	0.078	0.744	-	-	-
Eigen and Fergus [9]	795	0.158	-	0.641	0.769	0.950	0.988
Laina *et al.* [28]	96K	0.129	0.056	0.583	0.801	0.950	0.986
Li *et al.* [29]	96K	0.139	0.058	0.505	0.820	0.960	0.989
Xu *et al.* [54]	4.7K	0.139	0.063	0.609	0.793	0.948	0.984
Xu *et al.* [54]	95K	0.121	**0.052**	0.586	0.811	0.950	0.986
Joint HCRF [51]	795	0.220	0.094	0.745	0.605	0.890	0.970
Jafari *et al.* [18]	795	0.157	0.068	0.673	0.762	0.948	0.988
PAD-Net-ResNet50 (Ours)	795	**0.120**	0.055	**0.582**	**0.817**	**0.954**	**0.987**

Table 5. Quantitative comparison results with the state-of-the-art methods on the Cityscapes *test* set. Our model is trained only on the fine-annotation dataset.

Method	IoU cla.	iIoU cla.	IoU cat.	iIoU cat.
SegNet [2]	0.561	0.342	0.798	0.664
CRF-RNN [60]	0.625	0.344	0.827	0.660
SiCNN [24]	0.663	0.449	0.850	0.712
DPN [34]	0.668	0.391	0.860	0.691
Dilation10 [58]	0.671	0.420	0.865	0.711
LRR [11]	0.697	0.480	0.882	0.747
DeepLab [4]	0.704	0.426	0.864	0.677
Piecewise [31]	0.716	0.517	0.873	0.741
PSPNet [59]	0.784	0.567	0.906	0.786
PAD-Net-ResNet101 (Ours)	0.803	0.588	0.908	0.785

Table 6. Quantitative evaluation of the importance of intermediate supervision and multiple tasks.

Method	Depth Metrics			Parsing Metrics		
	rel	log10	rms	Mean IoU	Mean Acc	Pixel Acc
MTDN-mds	0.149	0.063	0.701	0.474	0.597	0.727
MTDN-inp0	0.153	0.069	0.721	0.465	0.588	0.713
MTDN-inp2	0.139	0.064	0.672	0.481	0.603	0.729
MTDN-inp3	0.128	0.059	0.617	0.490	0.612	0.739
MTDN-full	**0.120**	**0.055**	**0.582**	**0.502**	**0.623**	**0.752**

Effect of direct multi-task learning. To investigate the effect of simultaneously optimizing two different task as previous works [40, 51], *i.e.* predicting two different tasks directly from the last scale feature map of the front-end CNN. We carry out experiments on both the NYUD-v2 and Cityscapes datasets, as shown in Table. 1, 2 and Figure 5, respectively. It can be observed that on NYUD-v2, the Front-end + DE + SP slightly outperforms the Front-end + DE, while on Cityscapes, the performance of Front-end + DE + SP is even decreased, which means that using a direct multi-task learning as traditional is probably not an effective means to facilitate each other the performance of different tasks.

Effect of multi-modal distillation. We further evaluate the effect of the proposed three different distillation modules for incorporating information from different prediction tasks. Table 1 shows the results on the depth prediction task using PAD-Net embedded with the distillation module A, B and C. It can be seen that these three variants of PAD-Net are all obviously better than the two baseline methods, and

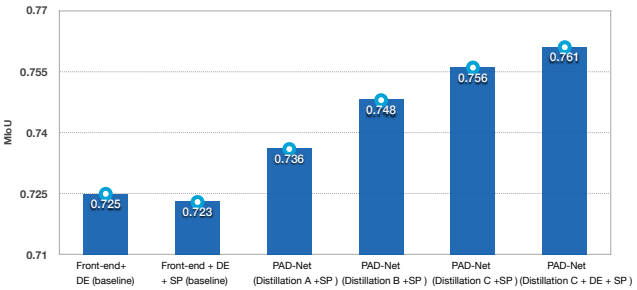

Figure 5. Diagnostic experiments of the proposed approach for the scene parsing on Cityscapes *val* dataset with ResNet-50 as the front-end backbone CNN.

the best one of ours, PAD-Net (Distillation C + DE) is 4.4 and 2.3 points better than the baseline Front-end + DE + SP on the rel and on the log10 metric respectively, and on the segmentation task on the same dataset, it is 3.1 points higher than the same baseline on the mIoU metric, which clearly demonstrates the effectiveness of the proposed multi-modal distillation strategy. Similar performance gaps can be also observed on the segmentation task on Cityscapes in Figure 5. For comparing the different distillation modules, the message passing between different tasks (the module B and C) significantly boosts the the performance over the naive combination method (the module C). By using the attention guided scheme, the performance of the module C is further improved over the module B.

Effect of multi-task guided simultaneous prediction. We finally verify that the proposed multi-tasks guided prediction and distillation approach facilitates boosting the performance of both the depth estimation and scene parsing. The results of PAD-Net (Distillation C + DE + SP) clearly outperforms PAD-Net (Distillation C + DE) and PAD-Net (Distillation C + SP) in both the depth estimation task (Table 1) and the segmentation task (Tabel 2 and Figure 5). This shows that our design of PAD-Net can use multiple final tasks in learning more effective features. More importantly, PAD-Net (Distillation C + DE + SP) obtains remarkably better performance than the baseline Front-end + DE + SP, further demonstrating the superiority of the proposed PAD-Net compared with the methods directly using two tasks to learn a deep network.

Importance of intermediate supervision and tasks. To evaluate the importance of the intermediate tasks, we use the multiple deep supervision, but consider different number of intermediate predictions for the distillation module, including MTDN-inp2 (2 inputs, depth + semantic map), MTDN-inp3 (3 inputs, depth + semantic map + surface normal) and MTDN-full (4 inputs). As shown in Table 6, MTDN-mds is obviously worse than MTDN-full, meaning that the performance gain is not only because of the model capacity. MTDN-inp0 is also worse than MTDN-full, showing that the improvement is not just because of

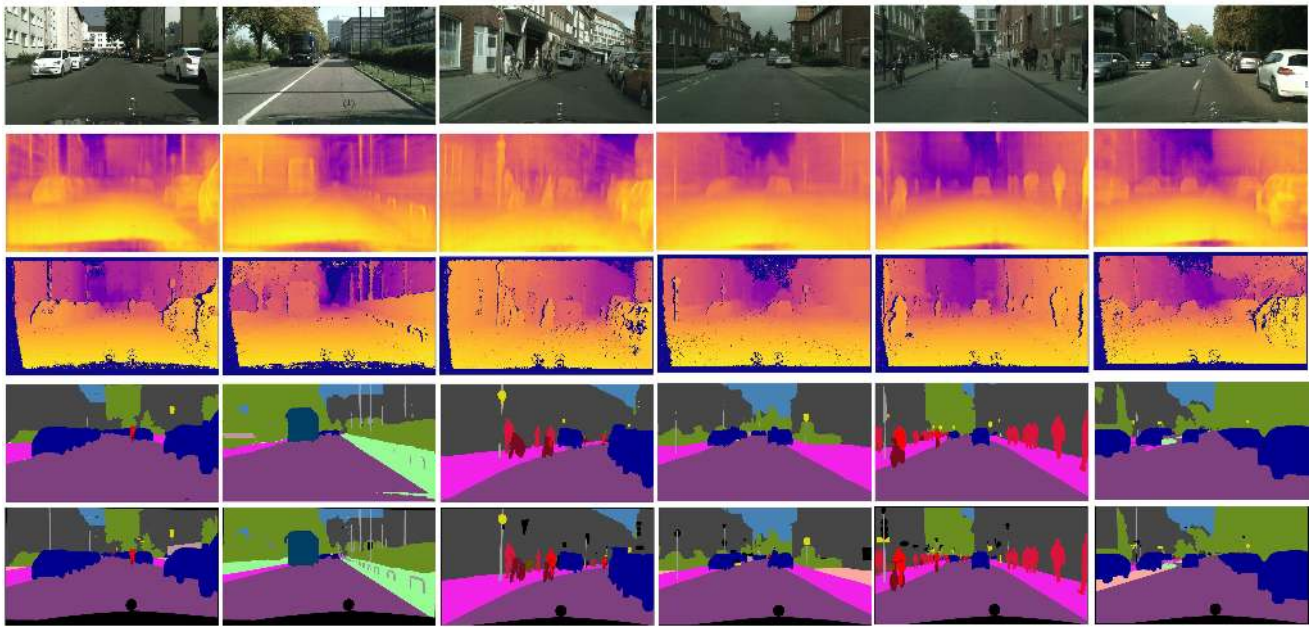

Figure 6. Quanlitative examples of depth prediction and 19-classes scene parsing results the Cityscapes dataset. The second and the fourth row correspond to the sparse depth and the semantic groundtruth, respectively.

adding the intermediate supervision. To evaluate the importance of the intermediate tasks, we use the multiple deep supervision, but consider different number of intermediate predictions for the distillation module, including MTDN-inp2 (2 inputs, depth + semantic map), MTDN-inp3 (3 inputs, depth + semantic map + surface normal) and MTDN-full (4 inputs). The results on NYUD-v2 are shown in Table 1. It is obvious that MTDN-mds is significantly worse than MTDN-full on both tasks (2.9% worse on rel, 2.8% worse on mIoU); Using more predictions as input gradually boosts the final performance: MTDN-full is 3.3% (on rel) and 3.7% (on mIoU) better than MTDN-inp0.

4.3. State-of-the-art Comparison

Depth estimation. On the depth estimation task, we compare with several state-of-the-art methods, including: methods adopting hand-crafted features and deep representations [46, 46, 20, 27, 10, 9, 29, 28], and methods considering graphical modeling with CNN [33, 32, 61, 51, 54]. As shown in Table 4, PAD-Net using ResNet-50 network as the front-end achieves the best performance in all the measure metrics among all the comparison methods. It should be noted that our approach is trained only on the official training set with 795 images without using extra training data. More importantly, to compare with the methods working on joint learning the two tasks (Joint HCRF [51] and Jafari *et al*. [18]), our performance is remarkably higher than theirs, further verifying the advantage of the proposed approach. As the Cityscapes dataset only provides the disparity map, we do not quantitatively evaluate the depth estimation performance on this dataset. Figure 4 and 6 show qualitative examples of the depth estimation on the two datasets.

Scene parsing. For the scene parsing task, we quantitatively compare the performance with the state of the art methods both on NYUD-v2 in Table 3 and on Cityscapes in Table 5. On NYUD-v2, our PAD-Net-ResNet50 significantly outperforms the runner up competitor RefineNet-Res152 [30] with a 3.7 points gap on the mIoU metric. On the cityscapes, we train ours only on the fine-annotation training set, ours achieves a class-level mIoU of 0.803, which is 1.9 points better than the best competitor PSPNet trained on the same set. Qualitative scene parsing examples are shown in Figure 4 and 6.

5. Conclusion

We have presented the proposed PAD-Net for simultaneous depth estimation and scene parsing. The PAD-Net introduces a novel deep multi-task learning means, which first predicts several intermediate auxiliary tasks and then employs the multi-task predictions as guidance to facilitate optimizing the final main tasks. Three different multi-modal distillation modules are developed to utilize the multi-task predictions more effectively. Our extensive experiments on NYUD-v2 and Cityscapes datasets demonstrated its effectiveness. We also provided new state of the art results on both the depth estimation and scene parsing tasks on NYUD-v2, and top performance on Cityscapes scene parsing task.

Acknowledgements

Wanli Ouyang is partially supported by SenseTime Group Limited. The authors would like to thank NVIDIA for GPU donation.

References

[1] A. Arnab, S. Jayasumana, S. Zheng, and P. H. Torr. Higher order conditional random fields in deep neural networks. In *ECCV*, 2016. 2

[2] V. Badrinarayanan, A. Handa, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015. 2, 7

[3] L.-C. 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. 3

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016. 1, 2, 7

[5] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016. 2

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 5

[7] E. Delage, H. Lee, and A. Y. Ng. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In *CVPR*, 2006. 2

[8] Z. Deng, S. Todorovic, and L. Jan Latecki. Semantic segmentation of rgbd images with mutex constraints. In *ICCV*, 2015. 6

[9] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 2, 6, 7, 8

[10] 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, 5, 7, 8

[11] G. Ghiasi and C. C. Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *ECCV*, 2016. 7

[12] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*, 2013. 2

[13] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*, 2014. 5, 6

[14] H. Han, A. K. Jain, S. Shan, and X. Chen. Heterogeneous face attribute estimation: A deep multi-task learning approach. *arXiv preprint arXiv:1706.00906*, 2017. 2

[15] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014. 2

[16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 2

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 3, 6

[18] O. H. Jafari, O. Groth, A. Kirillov, M. Y. Yang, and C. Rother. Analyzing modular cnn architectures for joint depth prediction and semantic segmentation. *arXiv preprint arXiv:1702.08009*, 2017. 1, 2, 7, 8

[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 5

[20] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *TPAMI*, 36(11):2144–2158, 2014. 7, 8

[21] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *arXiv preprint arXiv:1705.07115*, 2017. 1, 2

[22] I. Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017. 1

[23] S. Kong and C. Fowlkes. Recurrent scene parsing with perspective understanding in the loop. *arXiv preprint arXiv:1705.07238*, 2017. 6

[24] I. Krešo, D. Čaušević, J. Krapac, and S. Šegvić. Convolutional scale invariance for semantic segmentation. In *GCPR*. Springer, 2016. 7

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 3, 6

[26] R. Kuga, A. Kanezaki, M. Samejima, Y. Sugano, and Y. Matsushita. Multi-task learning using multi-modal encoder-decoder networks with shared skip connections. In *ICCVW*, 2017. 2

[27] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *CVPR*, 2014. 7, 8

[28] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. *arXiv preprint arXiv:1606.00373*, 2016. 2, 7, 8

[29] B. Li, Y. Dai, and M. He. Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference. *arXiv preprint arXiv:1708.02287*, 2017. 7, 8

[30] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation. *arXiv preprint arXiv:1611.06612*, 2016. 6, 8

[31] G. Lin, C. Shen, A. van den Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016. 6, 7

[32] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, 2015. 2, 5, 7, 8

[33] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *CVPR*, 2014. 2, 7, 8

[34] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015. 2, 7

[35] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2, 6

[36] M. Long and J. Wang. Learning multiple tasks with deep relationship networks. *arXiv preprint arXiv:1506.02117*, 2015. 2

[37] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015. 4

[38] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016. 2

[39] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *NIPS*, 2014. 4

[40] A. Mousavian, H. Pirsiavash, and J. Košecká. Joint semantic segmentation and depth estimation with deep convolutional networks. In *3DV*, 2016. 1, 2, 6, 7

[41] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015. 2

[42] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *arXiv preprint arXiv:1603.01249*, 2016. 1

[43] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *CVPR*, 2016. 2, 7

[44] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard. Sluice networks: Learning what to share between loosely related tasks. *arXiv preprint arXiv:1705.08142*, 2017. 2

[45] A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *IJCV*, 76(1):53–69, 2008. 2

[46] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *TPAMI*, 31(5):824–840, 2009. 7, 8

[47] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 5

[48] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 5

[49] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[50] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. *arXiv preprint arXiv:1612.07695*, 2016. 5

[51] 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. 1, 2, 7, 8

[52] F. Xia, P. Wang, L.-C. Chen, and A. L. Yuille. Zoom better to see clearer: Human and object parsing with hierarchical auto-zoom net. In *ECCV*, 2016. 2

[53] D. Xu, W. Ouyang, X. Alameda-Pineda, E. Ricci, X. Wang, and N. Sebe. Learning deep structured multi-scale features using attention-gated crfs for contour prediction. In *NIPS*, 2017. 2

[54] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *CVPR*, 2017. 1, 2, 5, 7, 8

[55] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *CVPR*, 2018. 2

[56] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 4

[57] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012. 2

[58] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 2, 3, 7

[59] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *arXiv preprint arXiv:1612.01105*, 2016. 7

[60] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 2, 7

[61] W. Zhuo, M. Salzmann, X. He, and M. Liu. Indoor scene structure analysis for single image depth estimation. In *CVPR*, 2015. 7, 8