Deep Ordinal Regression Network for Monocular Depth Estimation

Huan Fu[1] Mingming Gong[2,3] Chaohui Wang[4] Kayhan Batmanghelich[2] Dacheng Tao[1]
[1]UBTECH Sydney AI Centre, SIT, FEIT, The University of Sydney, Australia
[2]Department of Biomedical Informatics, University of Pittsburgh
[3]Department of Philosophy, Carnegie Mellon University
[4]Université Paris-Est, LIGM (UMR 8049), CNRS, ENPC, ESIEE Paris, UPEM, Marne-la-Vallée, France
{hufu6371@uni., dacheng.tao@}sydney.edu.au {mig73, kayhan@}pitt.edu chaohui.wang@u-pem.fr

Abstract

Monocular depth estimation, which plays a crucial role in understanding 3D scene geometry, is an ill-posed problem. Recent methods have gained significant improvement by exploring image-level information and hierarchical features from deep convolutional neural networks (DCNNs). These methods model depth estimation as a regression problem and train the regression networks by minimizing mean squared error, which suffers from slow convergence and unsatisfactory local solutions. Besides, existing depth estimation networks employ repeated spatial pooling operations, resulting in undesirable low-resolution feature maps. To obtain high-resolution depth maps, skip-connections or multi-layer deconvolution networks are required, which complicates network training and consumes much more computations. To eliminate or at least largely reduce these problems, we introduce a spacing-increasing discretization (SID) strategy to discretize depth and recast depth network learning as an ordinal regression problem. By training the network using an ordinary regression loss, our method achieves much higher accuracy and faster convergence in synch. Furthermore, we adopt a multi-scale network structure which avoids unnecessary spatial pooling and captures multi-scale information in parallel.

The method described in this paper achieves state-of-the-art results on four challenging benchmarks, i.e.*, KITTI [18], ScanNet [10], Make3D [51], and NYU Depth v2 [43], and win the 1st prize in Robust Vision Challenge 2018. Code has been made available at: `https://github.com/hufu6371/DORN`.*

1. Introduction

Estimating depth from 2D images is a crucial step of scene reconstruction and understanding tasks, such as 3D object recognition, segmentation, and detection. In this paper, we examine the problem of *Monocular Depth Estimation* from a single image (abbr. as *MDE* hereafter).

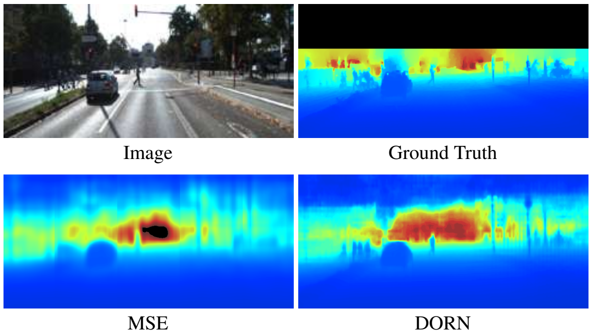

Figure 1: **Estimated Depth by DORN.** MSE: Training our network via MSE in log space, where ground truths are continuous depth values. DORN: The proposed deep ordinal regression network. Depth values in the black part are not provided by KITTI.

Compared to depth estimation from stereo images or video sequences, in which significant progresses have been made [21, 31, 28, 46], the progress of MDE is slow. MDE is an ill-posed problem: a single 2D image may be produced from an infinite number of distinct 3D scenes. To overcome this inherent ambiguity, typical methods resort to exploiting statistically meaningful monocular cues or features, such as perspective and texture information, object sizes, object locations, and occlusions [51, 26, 34, 50, 28].

Recently, some works have significantly improved the MDE performance with the use of DCNN-based models [40, 57, 48, 11, 30, 33, 35, 3], demonstrating that deep features are superior to handcrafted features. These methods address the MDE problem by learning a DCNN to estimate the continuous depth map. Since this problem is a standard regression problem, mean squared error (MSE) in log-space or its variants are usually adopted as the loss function. Although optimizing a regression network can achieve a reasonable solution, we find that the convergence is rather slow and the final solution is far from satisfactory.

In addition, existing depth estimation networks [11, 17, 33, 35, 40, 59] usually apply standard DCNNs designed initially for image classification in a full convolutional manner as the feature extractors. In these networks, repeated spatial pooling quickly reduce the spatial resolution of feature maps (usually stride of 32), which is undesirable for depth estimation. Though high-resolution depth maps can be obtained by incorporating higher-resolution feature maps via multi-layer deconvolutional networks [35, 17, 33], multi-scale networks [40, 11] or skip-connection [59], such a processing would not only require additional computational and memory costs, but also complicate the network architecture and the training procedure.

In contrast to existing developments for MDE, we propose to discretize continuous depth into a number of intervals and cast the depth network learning as an ordinal regression problem, and present how to involve ordinal regression into a dense prediction task via DCNNs. More specifically, we propose to perform the discretization using a spacing-increasing discretization (SID) strategy instead of the uniform discretization (UD) strategy, motivated by the fact that the uncertainty in depth prediction increases along with the underlying ground-truth depth, which indicates that it would be better to allow a relatively larger error when predicting a larger depth value to avoid over-strengthened influence of large depth values on the training process. After obtaining the discrete depth values, we train the network by an ordinal regression loss, which takes into account the ordering of discrete depth values.

To ease network training and save computational cost, we introduce a network architecture which avoids unnecessary subsampling and captures multi-scale information in a simpler way instead of skip-connections. Inspired by recent advances in scene parsing [62, 4, 6, 64], we first remove subsampling in the last few pooling layers and apply dilated convolutions to obtain large receptive fields. Then, multi-scale information is extracted from the last pooling layer by applying dilated convolution with multiple dilation rates. Finally, we develop a full-image encoder which captures image-level information efficiently at a significantly lower cost of memory than the fully-connected full-image encoders [2, 12, 11, 37, 30]. The whole network is trained in an end-to-end manner without stage-wise training or iterative refinement. Experiments on four challenging benchmarks, *i.e.*, KITTI [18], ScanNet [10], Make3D [51, 50] and NYU Depth v2 [43], demonstrate that the proposed method achieves state-of-the-art results, and outperforms recent algorithms by a significant margin.

The remainder of this paper is organized as follows. After a brief review of related literatures in Sec. 2, we present in Sec. 3 the proposed method in detail. In Sec. 4, besides the qualitative and quantitative performance on those benchmarks, we also evaluate multiple basic instantiations of the proposed method to analyze the effects of those core factors. Finally, we conclude the whole paper in Sec. 5.

2. Related Work

Depth Estimation is essential for understanding the 3D structure of scenes from 2D images. Early works focused on depth estimation from stereo images by developing geometry-based algorithms [52, 14, 13] that rely on point correspondences between images and triangulation to estimate the depth. In a seminal work [50], Saxena *et al.* learned the depth from monocular cues in 2D images via supervised learning. Since then, a variety of approaches have been proposed to exploit the monocular cues using handcrafted representations [51, 26, 34, 38, 8, 32, 1, 55, 47, 16, 22, 61]. Since handcrafted features alone can only capture local information, probabilistic graphic models such as Markov Random Fields (MRFs) are often built based on these features to incorporate long-range and global cues [51, 65, 41]. Another successful way to make use of global cues is the *DepthTransfer* method [28] which uses GIST global scene features [45] to search for candidate images that are “similar” to the input image from a database containing RGBD images.

Given the success of DCNNs in image understanding, many depth estimation networks have been proposed in recent years [20, 63, 37, 42, 54, 58, 48, 40, 29]. Thanks to multi-level contextual and structural information from powerful very deep networks (*e.g.*, *VGG* [56] and *ResNet* [24]), depth estimation has been boosted to a new accuracy level [11, 17, 33, 35, 59]. The main hurdle is that the repeated pooling operations in these deep feature extractors quickly decrease the spatial resolution of feature maps (usually stride 32). Eigen *et al.* [12, 11] applied multi-scale networks which stage-wisely refine estimated depth map from low spatial resolution to high spatial resolution via independent networks. Xie *et al.* [59] adopted the skip-connection strategy to fuse low-spatial resolution depth map in deeper layers with high-spatial resolution depth map in lower layers. More recent works [17, 33, 35] apply multi-layer deconvolutional networks to recover coarse-to-fine depth. Rather than solely relying on deep networks, some methods incorporate conditional random fields to further improve the quality of estimated depth maps [57, 40]. To improve efficiency, Roy and Todorovic [48] proposed the Neural Regression Forest method which allows for parallelizable training of “shallow” CNNs.

Recently, unsupervised or semi-supervised learning is introduced to learn depth estimation networks [17, 33]. These methods design reconstruction losses to estimate the disparity map by recovering a right view with a left view. Also, some weakly-supervised methods considering

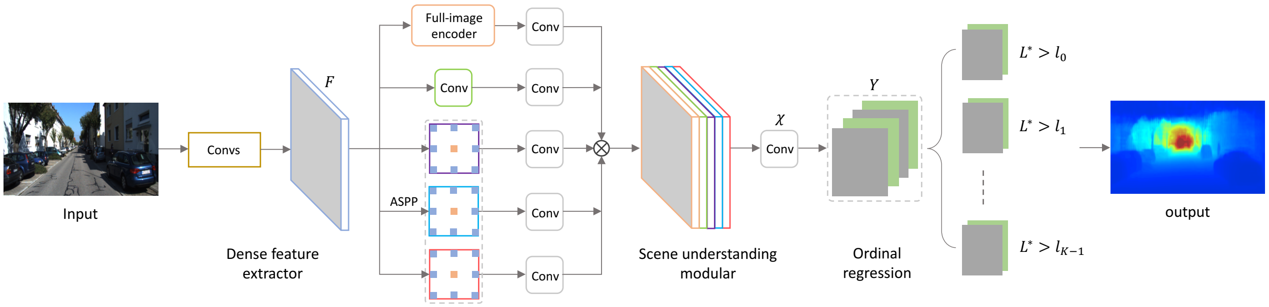

Figure 2: **Illustration of the network architecture.** The network consists of a dense feature extractor, multi-scale feature learner (*ASPP*), cross channel information learner (the pure 1×1 convolutional branch), a full-image encoder and an ordinal regression optimizer. The *Conv* components here are all with kernel size of 1×1. The *ASPP* module consists of 3 dilated convolutional layers with kernel size of 3×3 and dilated rate of 6, 12 and 18 respectively [6]. The supervised information of our network is discrete depth values output by the discretization using the *SID* strategy. The whole network is optimized by our ordinal regression training loss in an end-to-end fashion.

pair-wise ranking information were proposed to roughly estimate and compare depth [66, 7].

Ordinal Regression [25, 23] aims to learn a rule to predict labels from an ordinal scale. Most literatures modify well-studied classification algorithms to address ordinal regression algorithms. For example, Shashua and Levin [53] handled multiple thresholds by developing a new SVM. Cammer and Singer [9] generalized the online perceptron algorithms with multiple thresholds to do ordinal regression. Another way is to formulate ordinal regression as a set of binary classification subproblems. For instance, Frank and Hall [15] applied some decision trees as binary classifiers for ordinal regression. In computer vision, ordinal regression has been combined with DCNNs to address the age estimation problem [44].

3. Method

This section first introduces the architecture of our deep ordinal regression network; then presents the SID strategy to divide continuous depth values into discrete values; and finally details how the network parameters can be learned in the ordinal regression framework.

3.1. Network Architecture

As shown in Fig. 2, the divised network consists of two parts, *i.e.*, a dense feature extractor and a scene understanding modular, and outputs multi-channel dense ordinal labels given an image.

3.1.1 Dense Feature Extractor

Previous depth estimation networks [11, 17, 33, 35, 40, 59] usually apply standard DCNNs originally designed for image recognition as the feature extractor. However, the repeated combination of max-pooling and striding significantly reduces the spatial resolution of the feature maps. Also, to incorporate multi-scale information and reconstruct high-resolution depth maps, some partial remedies, including stage-wise refinement [12, 11], skip connection [59] and multi-layer deconvolution network [17, 33, 35] can be adopted, which nevertheless not only requires additional computational and memory cost, but also complicates the network architecture and the training procedure. Following some recent scene parsing network [62, 4, 6, 64], we advocate removing the last few downsampling operators of DCNNs and inserting holes to filters in the subsequent *conv* layers, called dilated convolution, to enlarge the field-of-view of filters without decreasing spatial resolution or increasing number of parameters.

3.1.2 Scene Understanding Modular

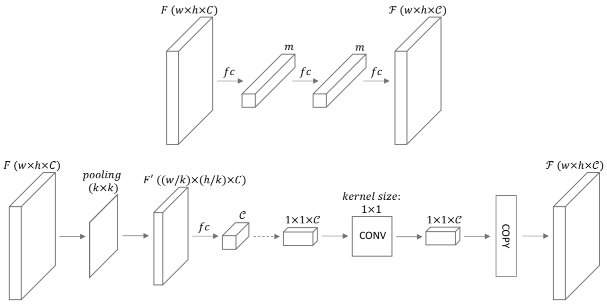
$F\ (w \times h \times C)$
m
m
fc
fc
fc
$\mathcal{F}\ (w \times h \times \mathcal{C})$
$F\ (w \times h \times C)$
pooling $(k \times k)$
$F'\ ((w/k) \times (h/k) \times C)$
fc
$\mathcal{C}$
$1 \times 1 \times \mathcal{C}$
kernel size: 1×1
CONV
$1 \times 1 \times \mathcal{C}$
COPY
$\mathcal{F}\ (w \times h \times \mathcal{C})$

Figure 3: **Full-Image Encoders**. Top: the full-image encoder implemented by pure fc layers [12, 11, 2] ($\delta < 1.25$: 0.910); Bottom: Our proposed encoder ($\delta < 1.25$: 0.915).

相当于对最高层信息进行融合，得到1x1xc的特征向量(描述场景信息)，然后再decoder得到深度图

The scene understanding modular consists of three parallel components, *i.e.*, an atrous spatial pyramid pooling

(*ASPP*) module [5, 6], a cross-channel leaner, and a full-image encoder. *ASPP* is employed to extract features from multiple large receptive fields via dilated convolutional operations. The dilation rates are 6, 12 and 18, respectively. The pure 1×1 convolutional branch can learn complex cross-channel interactions. The full-image encoder captures global contextual information and can greatly clarify local confusions in depth estimation [57, 12, 11, 2].

Though previous methods have incorporated full-image encoders, our full-image encoder contains fewer parameters. As shown in Fig. 3, to obtain global feature $\mathcal{F}$ with dimension $\mathcal{C} \times h \times w$ from F with dimension $C \times h \times w$, a common *fc*-fashion method accomplishes this by using fully-connected layers, where each element in $\mathcal{F}$ connects to all the image features, implying a global understanding of the entire image. However, this method contains a prohibitively large number of parameters, which is difficult to train and is memory consuming. In contrast, we first make use of an average pooling layer with a small kernel size and stride to reduce the spatial dimensions, followed by a fc layer to obtain a feature vector with dimension $\mathcal{C}$. Then, we treat the feature vector as $\mathcal{C}$ channels of feature maps with spatial dimensions of 1×1, and add a *conv* layer with the kernel size of 1×1 as a cross-channel parametric pooling structure. Finally, we copy the feature vector to $\mathcal{F}$ along spatial dimensions so that each location of $\mathcal{F}$ share the same understanding of the entire image.

The obtained features from the aforementioned components are concatenated to achieve a comprehensive understanding of the input image. Also, we add two additional convolutional layers with the kernel size of 1×1, where the former one reduces the feature dimension and learns complex cross-channel interactions, and the later one transforms the features into multi-channel dense ordinal labels.

3.2. Spacing-Increasing Discretization

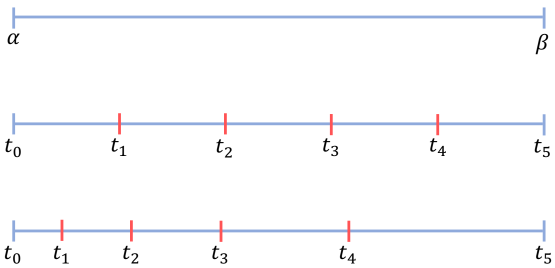

Figure 4: **Discrete Intervals.** Illustration of UD (middle) and SID (bottom) to discretize depth interval $[\alpha, \beta]$ into five sub-intervals. See Eq. 1 for details.

To quantize a depth interval $[\alpha, \beta]$ into a set of representative discrete values, a common way is the uniform discretization (UD). However, as the depth value becomes larger, the information for depth estimation is less rich, meaning that the estimation error of larger depth values is generally larger. Hence, using the UD strategy would induce an over-strengthened loss for the large depth values. To this end, we propose to perform the discretization using the SID strategy (as shown in Fig. 4), which uniformed discretizes a given depth interval in *log* space to down-weight the training losses in regions with large depth values, so that our depth estimation network is capable to more accurately predict relatively small and medium depth and to rationally estimate large depth values. Assuming that a depth interval $[\alpha, \beta]$ needs to be discretized into K sub-intervals, UD and SID can be formulated as:

$$\begin{aligned} \text{UD:} \quad & t_i = \alpha + (\beta - \alpha) * i/K, \\ \text{SID:} \quad & t_i = e^{\log(\alpha) + \frac{\log(\beta/\alpha) * i}{K}}, \end{aligned} \tag{1}$$

where $t_i \in \{t_0, t_1, ..., t_K\}$ are discretization thresholds. In our paper, we add a shift ξ to both α and β to obtain α^* and β^* so that $\alpha^* = \alpha + \xi = 1.0$, and apply SID on $[\alpha^*, \beta^*]$.

3.3. Learning and Inference

After obtaining the discrete depth values, it is straightforward to turn the standard regression problem into a multi-class classification problem, and adopts *softmax* regression loss to learn the parameters in our depth estimation network. However, typical multi-class classification losses ignore the ordered information between the discrete labels, while depth values have a strong ordinal correlation since they form a well-ordered set. Thus, we cast the depth estimation problem as an ordinal regression problem and develop an ordinal loss to learn our network parameters.

Let $\chi = \varphi(I, \Phi)$ denote the feature maps of size $W \times H \times C$ given an image I, where Φ is the parameters involved in the dense feature extractor and the scene understanding modular. $Y = \psi(\chi, \Theta)$ of size $W \times H \times 2K$ denotes the ordinal outputs for each spatial locations, where $\Theta = (\theta_0, \theta_1, ..., \theta_{2K-1})$ contains weight vectors. And $l_{(w,h)} \in \{0, 1, ..., K-1\}$ is the discrete label produced by SID at spatial location (w, h). Our ordinal loss $\mathcal{L}(\chi, \Theta)$ is defined as the average of pixelwise ordinal loss $\Psi(h, w, \chi, \Theta)$ over the entire image domain:

$$\begin{aligned} \mathcal{L}(\chi, \Theta) &= -\frac{1}{\mathcal{N}} \sum_{w=0}^{W-1} \sum_{h=0}^{H-1} \Psi(w, h, \chi, \Theta), \\ \Psi(h, w, \chi, \Theta) &= \sum_{k=0}^{l_{(w,h)}-1} \log(\mathcal{P}_{(w,h)}^{k}) \\ &\quad + \sum_{k=l_{(w,h)}}^{K-1} (\log(1 - \mathcal{P}_{(w,h)}^{k})), \\ \mathcal{P}_{(w,h)}^{k} &= P(\hat{l}_{(w,h)} > k | \chi, \Theta), \end{aligned} \tag{2}$$

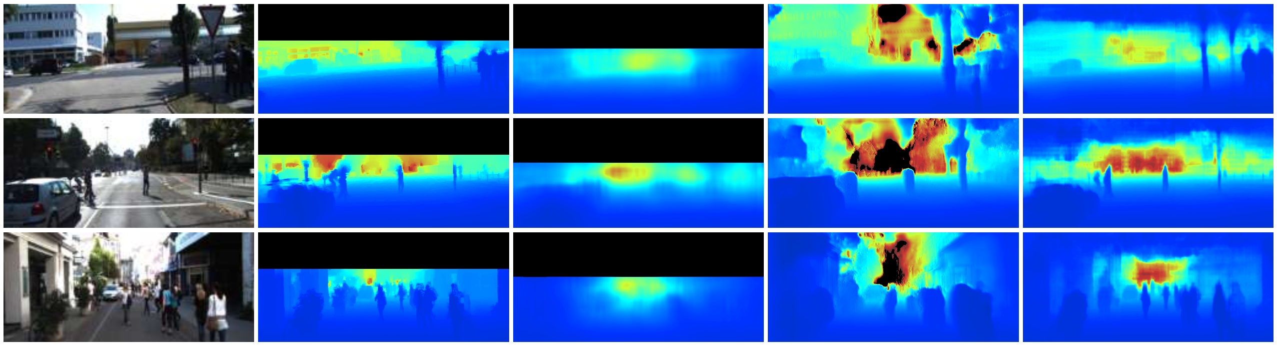
Image | Ground Truth | Eigen [11] | LRC [18] | DORN

Figure 5: **Depth Prediction on KITTI.** Image, ground truth, Eigen [12], LRC [19], and our DORN. Ground truth has been interpolated for visualization. Pixels with distance $> 80m$ in LRC are masked out.

where $\mathcal{N} = W \times H$, and $\hat{l}_{(w,h)}$ is the estimated discrete value decoding from $y_{(w,h)}$. We choose *softmax* function to compute $\mathcal{P}^k_{(w,h)}$ from $y_{(w,h,2k)}$ and $y_{(w,h,2k+1)}$ as follows:

$$\mathcal{P}^k_{(w,h)} = \frac{e^{y_{(w,h,2k+1)}}}{e^{y_{(w,h,2k)}} + e^{y_{(w,h,2k+1)}}}, \tag{3}$$

where $y_{(w,h,i)} = \theta_i^T x_{(w,h)}$, and $x_{(w,h)} \in \chi$. Minimizing $\mathcal{L}(\chi, \Theta)$ ensures that predictions farther from the true label incur a greater penalty than those closer to the true label.

The minimization of $\mathcal{L}(\chi, \Theta)$ can be done via an iterative optimization algorithm. Taking derivate with respect to θ_i, the gradient takes the following form:

$$\begin{aligned}
\frac{\partial \mathcal{L}(\chi, \Theta)}{\partial \theta_i} &= -\frac{1}{\mathcal{N}} \sum_{w=0}^{W-1} \sum_{h=0}^{H-1} \frac{\partial \Psi(w, h, \chi, \Theta)}{\partial \theta_i}, \\
\frac{\partial \Psi(w, h, \chi, \Theta)}{\partial \theta_{2k+1}} &= -\frac{\partial \Psi(w, h, \chi, \Theta)}{\partial \theta_{2k}}, \\
\frac{\partial \Psi(w, h, \chi, \Theta)}{\partial \theta_{2k}} &= x_{(w,h)} \eta(l_{(w,h)} > k)(\mathcal{P}^k_{(w,h)} - 1) \\
&\quad + x_{(w,h)} \eta(l_{(w,h)} \leq k) \mathcal{P}^k_{(w,h)},
\end{aligned} \tag{4}$$

where $k \in \{0, 1, ..., K-1\}$, and $\eta(\cdot)$ is an indicator function such that η(true) = 1 and η(false) = 0. We the can optimize our network via backpropagation.

In the inference phase, after obtaining ordinal labels for each position of image I, the predicted depth value $\hat{d}_{(w,h)}$ is decoded as:

$$\begin{aligned}
\hat{d}_{(w,h)} &= \frac{t_{\hat{l}_{(w,h)}} + t_{\hat{l}_{(w,h)}+1}}{2} - \xi, \\
\hat{l}_{(w,h)} &= \sum_{k=0}^{K-1} \eta(\mathcal{P}^k_{(w,h)} >= 0.5).
\end{aligned} \tag{5}$$

4. Experiments

To demonstrate the effectiveness of our depth estimator, we present a number of experiments examining different aspects of our approach. After introducing the implementation details, we evaluate our methods on three challenging outdoor datasets, *i.e. KITTI* [18], *Make3D* [50, 51] and NYU Depth v2 [43]. The evaluation metrics are following previous works [12, 40]. Some ablation studies based on *KITTI* are discussed to give a more detailed analysis of our method.

Implementation Details We implement our depth estimation network based on the public deep learning platform *Caffe* [27]. The learning strategy applies a polynomial decay with a base learning rate of 0.0001 and the power of 0.9. Momentum and weight decay are set to 0.9 and 0.0005 respectively. The iteration number is set to 300K for KITTI, 50K for Make3D, and 3M for NYU Depth v2, and batch size is set to 3. We find that further increasing the iteration number can only slightly improve the performance. We adopt both *VGG-16* [56] and *ResNet-101* [24] as our feature extractors, and initialize their parameters via the pre-trained classification model on ILSVRC [49]. Since features in first few layers only contain general low-level information, we fixed the parameters of *conv1* and *conv2* blocks in *ResNet* after initialization. Also, the batch normalization parameters in *ResNet* are directly initialized and fixed during training progress. Data augmentation strategies are following [12]. In the test phase, we split each image to some overlapping windows according the cropping method in the training phase, and obtain the predicted depth values in overlapped regions by averaging the predictions.

4.1. Benchmark Perfomance

KITTI The KITTI dataset [18] contains outdoor scenes with images of resolution about 375 × 1241 captured by

Method	abs rel.	imae	irmse	log mae	log rmse	mae	rmse	scale invar.	sq. rel.
Official Baseline	0.25	0.17	0.21	0.24	0.29	0.42	0.53	0.05	0.14
DORN	**0.14**	**0.10**	**0.13**	**0.13**	**0.17**	**0.22**	**0.29**	**0.02**	**0.06**

Table 1: **Scores on the online ScanNet evaluation server.** See https://goo.gl/8keUQN.

Method	SILog	sqErrorRel	absErrorRel	iRMSE
Official Baseline	18.19	7.32	14.24	18.50
DORN	**11.77**	**2.23**	**8.78**	**12.98**

Table 2: **Scores on the online KITTI evaluation server.** See https://goo.gl/iXuhiN.

cameras and depth sensors in a driving car. All the 61 scenes from the "city", "residential", "road" and "Campus" categories are used as our training/test sets. We test on 697 images from 29 scenes split by Eigen *et al.* [12], and train on about 23488 images from the remaining 32 scenes. We train our model on a random crop of size 385 × 513. For some other details, we set the maximal ordinal label for KITTI as 80, and evaluate our results on a pre-defined center cropping following [12] with the depth ranging from $0m$ to $80m$ and $0m$ to $50m$. Note that, a single model is trained on the full depth range, and is tested on data with different depth ranges.

Make3D The Make3D dataset [50, 51] contains 534 outdoor images, 400 for training, and 134 for testing, with the resolution of 2272 × 1704, and provides the ground truth depth map with a small resolution of 55 × 305. We reduce the resolution of all images to 568 × 426, and train our model on a random crop of size 513 × 385. Following previous works, we report *C1* (depth range from $0m$ to $80m$) and *C2* (depth range from $0m$ to $70m$) error on this dataset using three commonly used evaluation metrics [28, 40]. For the VGG model, we train our DORN on a depth range of $0m$ to $80m$ from scratch (ImageNet model), and evaluate results using the same model for *C1* and *C2* . However, for ResNet, we learn two separate models for $C1$ and $C2$ respectively.

NYU Depth v2 The NYU Depth v2 [43] dataset contains 464 indoor video scenes taken with a Microsoft Kinect camera. We train our DORN using all images (about 120K) from the 249 training scenes, and test on the 694-image test set following previous works. To speed up training, all the images are reduced to the resolution of 288 × 384 from 480 × 640. And the model are trained on random crops of size 257 × 353. We report our scores on a pre-defined center cropping by Eigen [12].

ScanNet The ScanNet [10] dataset is also a challenging benchmark which contains various indoor scenes. We train our model on the officially provided 24353 training and validation images with a random crop size of 385 × 513, and evaluate our method on the ScanNet online test server.

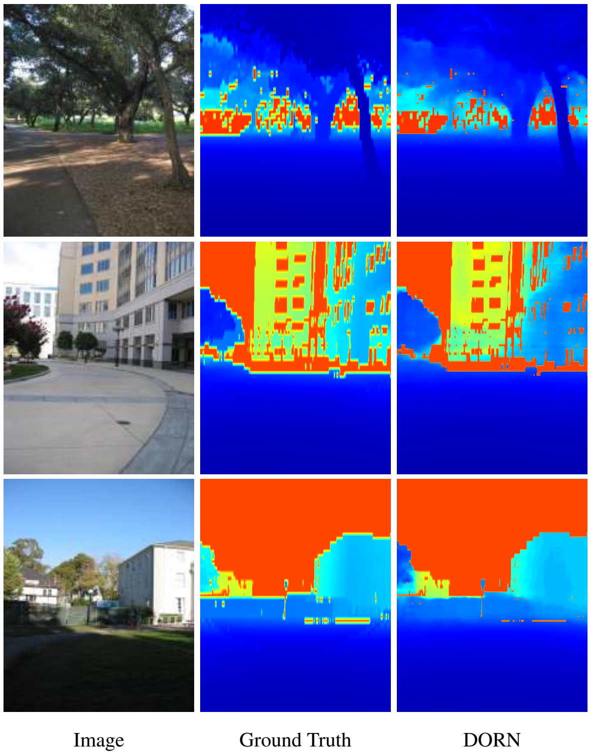

Figure 6: **Depth Prediction on Make3D.** Image, ground truth, and our DORN. Pixels with distance $> 70m$ are masked out.

Performance Tab. 3 and Tab. 4 give the results on two outdoor datasets, i.e., KITTI and Make3D. It can be seen that our DORN improves the accuracy by $5\% \sim 30\%$ in terms of all metrics compared with previous works in all settings. Some qualitative results are shown in Fig. 5 and Fig. 6. In Tab. 5, our DORN outperforms other methods on NYU Depth v2, which is one of the largest indoor benchmarks. The results suggest that our method is applicable to both indoor and outdoor data. We evaluate our method on the online KITTI evaluation server and the online ScanNet evaluation server. As shown in Tab. 2 and 1, our DORN significantly outperforms the officially provided baselines.

4.2. Ablation Studies

We conduct various ablation studies to analyze the details of our approach. Results are shown in Tab. 6, Tab. 7,

Method	cap	higher is better			lower is better			
		$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Abs Rel	Squa Rel	RMSE	RMSE$_{log}$
Make3D [51]	0 - 80 m	0.601	0.820	0.926	0.280	3.012	8.734	0.361
Eigen *et al.* [12]	0 - 80 m	0.692	0.899	0.967	0.190	1.515	7.156	0.270
Liu *et al.* [40]	0 - 80 m	0.647	0.882	0.961	0.217	1.841	6.986	0.289
LRC (CS + K) [19]	0 - 80 m	0.861	0.949	0.976	0.114	0.898	4.935	0.206
Kuznietsov *et al.* [33]	0 - 80 m	0.862	0.960	0.986	0.113	0.741	4.621	0.189
DORN (VGG)	0 - 80 m	0.915	0.980	0.993	0.081	0.376	3.056	0.132
DORN (ResNet)	0 - 80 m	**0.932**	**0.984**	**0.994**	**0.072**	**0.307**	**2.727**	**0.120**
Garg *et al.* [17]	0 - 50 m	0.740	0.904	0.962	0.169	1.080	5.104	0.273
LRC (CS + K) [19]	0 - 50 m	0.873	0.954	0.979	0.108	0.657	3.729	0.194
Kuznietsov *et al.* [33]	0 - 50 m	0.875	0.964	0.988	0.108	0.595	3.518	0.179
DORN (VGG)	0 - 50 m	0.920	0.982	0.994	0.079	0.324	2.517	0.128
DORN (ResNet)	0 - 50 m	**0.936**	**0.985**	**0.995**	**0.071**	**0.268**	**2.271**	**0.116**

Table 3: **Performance on KITTI.** All the methods are evaluated on the test split by Eigen *et al.* [12]. LRC (CS + K): LRC pre-train their model on Cityscapes and fine tune on KITTI.

Method	C1 error			C2 error		
	rel	log$_{10}$	rms	rel	log$_{10}$	rms
Make3D [51]	-	-	-	0.370	0.187	-
Liu *et al.* [39]	-	-	-	0.379	0.148	-
DepthTransfer [28]	0.355	0.127	9.20	0.361	0.148	15.10
Liu *et al.* [41]	0.335	0.137	9.49	0.338	0.134	12.60
Li *et al.* [36]	0.278	0.092	7.12	0.279	0.102	10.27
Liu *et al.* [40]	0.287	0.109	7.36	0.287	0.122	14.09
Roy *et al.* [48]	-	-	-	0.260	0.119	12.40
Laina *et al.* [35]	0.176	0.072	4.46	-	-	-
LRC-Deep3D [59]	1.000	2.527	19.11	-	-	-
LRC [19]	0.443	0.156	11.513	-	-	-
Kuznietsov *et al.* [33]	0.421	0.190	8.24	-	-	-
MS-CRF [60]	0.184	0.065	4.38	0.198	-	8.56
DORN (VGG)	0.236	0.082	7.02	0.238	0.087	10.01
DORN (ResNet)	**0.157**	**0.062**	**3.97**	**0.162**	**0.067**	**7.32**

Table 4: **Performance on Make3D.** LRC-Deep3D [59] is adopting LRC [19] on Deep3D model [59].

Method	δ_1	δ_2	δ_3	rel	log$_{10}$	rms
Make3D [51]	0.447	0.745	0.897	0.349	-	1.214
DepthTransfer [28]	-	-	-	0.35	0.131	1.2
Liu *et al.* [41]	-	-	-	0.335	0.127	1.06
Ladicky *et al.* [34]	0.542	0.829	0.941	-	-	-
Li *et al.* [36]	0.621	0.886	0.968	0.232	0.094	0.821
Wang *et al.* [57]	0.605	0.890	0.970	0.220	-	0.824
Roy *et al.* [48]	-	-	-	0.187	-	0.744
Liu *et al.* [40]	0.650	0.906	0.976	0.213	0.087	0.759
Eigen *et al.* [11]	0.769	0.950	0.988	0.158	-	0.641
Chakrabarti *et al.* [2]	0.806	0.958	0.987	0.149	-	0.620
Laina *et al.* [35]	0.629	0.889	0.971	0.194	0.083	0.790
Li *et al.* [37]	0.789	0.955	0.988	0.152	0.064	0.611
Laina *et al.* [35]†	0.811	0.953	0.988	0.127	0.055	0.573
Li *et al.* [37]†	0.788	0.958	0.991	0.143	0.063	0.635
MS-CRF [60]†	0.811	0.954	0.987	0.121	0.052	0.586
DORN†	**0.828**	**0.965**	**0.992**	**0.115**	**0.051**	**0.509**

Table 5: **Performance on NYU Depth v2.** δ_i: $\delta < 1.25^i$. †: ResNet based model.

Fig. 1, and Fig. 7, and discussed in detail.

4.2.1 Depth Discretization

Depth discretization is critical to performance improvement, because it allows us to apply classification and ordinal regression losses to optimize the network parameters. According to scores in Tab. 6, training by regression on continuous depth seems to converge to a poorer solution than the other two methods, and our ordinal regression network achieves the best performance. There is an obvious gap between approaches where depth is discretized by SID and UD, respectively. Besides, when replacing our ordinal regression loss by an advantage regression loss (*i.e.* BerHu), our DORN still obtain much higher scores. Thus, we can conclude that: (i) SID is important and can further improve the performance compared to UD; (i) discretizing depth and training using a multi-class classification loss is better than training using regression losses; (iii) exploring the ordinal correlation among depth drives depth estimation networks to converge to even better solutions.

Furthermore, we also train the network using RMSE$_{log}$ on discrete depth values obtained by SID, and report the results in Tab. 6. We can see that MSE-SID performs slightly better than MSE, which demonstrates that quantization errors are nearly ignorable in depth estimation. The benefits of discretization through the use of ordinal regression losses far exceeds the cost of depth discretization.

Variant	Iteration	higher is better			lower is better			
		$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Abs Rel	Squa Rel	RMSE	RMSE$_{log}$
MSE	1M	0.864	0.969	0.991	0.109	0.527	3.660	0.164
MSE-SID	0.6M	0.865	0.970	0.992	0.108	0.520	3.636	0.163
MCC-UD	0.3M	0.892	0.970	0.988	0.093	0.474	3.438	0.155
MCC-SID	0.3M	0.906	0.976	0.991	0.084	0.417	3.201	0.142
DORN-UD	0.3M	0.900	0.973	0.991	0.091	0.452	3.339	0.148
DORN-SID	0.3M	**0.915**	**0.980**	**0.993**	**0.081**	**0.376**	**3.056**	**0.132**
berHu†	0.6M	0.909	0.978	0.992	0.086	0.385	3.365	0.136
DORN†	0.3M	**0.932**	**0.984**	**0.994**	**0.072**	**0.307**	**2.727**	**0.120**

Table 6: **Depth Discretization and Ordinal Regression.** MSE: mean squared error in log space. MCC: multi-class classification. DORN: proposed ordinal regression. Note that training by MSE for 1M iterations only slightly improve the performance compared with 0.5M (about 0.001 on $\delta < 1.25$). berHu: the reverse Huber loss. †: ResNet based model.

4.2.2 Full-image Encoder

Variant	$\delta < 1.25$	Abs Rel	RMSE$_{log}$	Params
w/o full-image encoder	0.906	0.092	0.143	0M
fc-fashion	0.910	0.085	0.137	$753M$
our encoder	0.915	0.081	0.132	$51M$

Table 7: **Full-image Encoder.** Parameters here is computed by some common settings in Eigen [12] and our DORN.

From Tab. 7, a full-image encoder is important to further boost the performance. Our full-image encoder yields a little higher scores than *fc* type encoders [2, 12, 11, 37, 30], but significantly reduce the number of parameters. For example, we set C to 512 (VGG), $\mathcal{C}$ to 512, m to 2048 (Eigen [12, 11]), and k to 4 in Fig. 3. Because of limited computation resources, when implementing the *fc*-fashion encoder, we downsampled the resolution of F using the stride of 3, and upsampled $\mathcal{F}$ to the required resolution. With an input image of size 385×513, h and w will be 49 and 65 respectively in our network. The number of parameters in fc-fashion encoder and our encoder is $\frac{1}{9} * m * w * h * C + m^2 + \frac{1}{9} * w * h * \mathcal{C} * m \approx 753M$, and is $\mathcal{C} * \frac{w}{4} * \frac{h}{4} * C + \mathcal{C} * \mathcal{C} \approx 51M$, respectively. From the experimental results and parameter analysis, it can be seen that our full-image encoder performs better while requires less computational resources.

4.2.3 How Many Intervals

To illustrate the sensitivity to the number of intervals, we discretizing depth into various number of intervals via SID. As shown in Fig. 7, with a range of 40 to 120 intervals, our DORN has a score in $[0.908, 0.915]$ regarding $\delta < 1.25$, and a score in $[3.056, 3.125]$ in terms of RMSE, and is thereby robust to a long range of depth interval numbers. We can also see that neither too few nor too many depth intervals are rational for depth estimation: too few depth intervals cause large quantization error, while too many depth intervals lose the advantage of discretization.

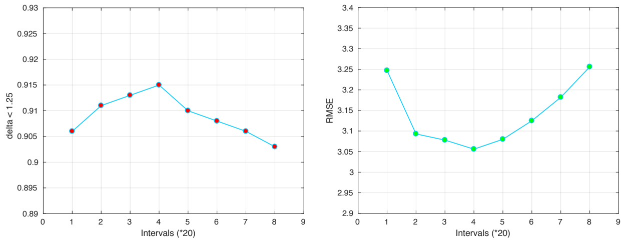

Figure 7: **Performance Ranging Different Intervals via SID.** Left: accuracy on $\delta < 1.25$. Right: evaluation errors on RMSE.

5. Conclusion

In this paper, we have developed an deep ordinal regression network (DORN) for monocular depth estimation MDE from a single image, consisting of a clean CNN architecture and some effective strategies for network optimization. Our method is motivated by two aspects: (i) to obtain high-resolution depth map, previous depth estimation networks require incorporating multi-scale features as well as full-image features in a complex architecture, which complicates network training and largely increases the computational cost; (ii) training a regression network for depth estimation suffers from slow convergence and unsatisfactory local solutions. To this end, we first introduced a simple depth estimation network which takes advantage of dilated convolution technique and a novel full-image encoder to directly obtain a high-resolution depth map. Moreover, an effective depth discretization strategy and an ordinal regression training loss were intergrated to improve the training of our network so as to largely increase the estimation accuracy. The proposed method achieves the state-of-the-art performance on the KITTI, ScanNet, Make3D and NYU

Depth v2 datasets. In the future, we will investigate new approximations to depth and extend our framework to other dense prediction problems.

6. Acknowledgement

This research was supported by Australian Research Council Projects FL-170100117 and DP-180103424. This work was partially supported by SAP SE and CNRS INS2I-JCJC-INVISANA. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research. This research was partially supported by research grant from Pfizer titled ”Developing Statistical Method to Jointly Model Genotype and High Dimensional Imaging Endophenotype.” We were also grateful for the computational resources provided by Pittsburgh Super Computing grant number TG-ASC170024.

References

[1] M. H. Baig and L. Torresani. Coupled depth learning. In *WACV*, 2016. 2

[2] A. Chakrabarti, J. Shao, and G. Shakhnarovich. Depth from a single image by harmonizing overcomplete local network predictions. In *NIPS*, 2016. 2, 3, 4, 7

[3] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *3DV*, 2017. 1

[4] 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. 2, 3

[5] 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:1606.00915*, 2016. 3

[6] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2, 3

[7] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *NIPS*, 2016. 3

[8] S. Choi, D. Min, B. Ham, Y. Kim, C. Oh, and K. Sohn. Depth analogy: Data-driven approach for single image depth estimation using gradient samples. *IEEE TIP*, 24(12):5953–5966, 2015. 2

[9] K. Crammer and Y. Singer. Pranking with ranking. In *NIPS*, 2002. 3

[10] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 1, 2, 6

[11] 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, 4, 7, 8

[12] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014. 2, 3, 4, 5, 6, 7, 8

[13] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *CVPR*, 2016. 2

[14] D. Forsyth and J. Ponce. *Computer Vision: a Modern Approach*. Prentice Hall, 2002. 2

[15] E. Frank and M. Hall. A simple approach to ordinal classification. *ECML*, 2001. 3

[16] R. Furukawa, R. Sagawa, and H. Kawasaki. Depth estimation using structured light flow – analysis of projected pattern flow on an object's surface. In *ICCV*, 2017. 2

[17] R. Garg, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016. 2, 3, 7

[18] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013. 1, 2, 5

[19] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *CVPR*, 2017. 5, 7

[20] R. A. Güler, G. Trigeorgis, E. Antonakos, P. Snape, S. Zafeiriou, and I. Kokkinos. Densereg: Fully convolutional dense shape regression in-the-wild. In *CVPR*, 2016. 2

[21] H. Ha, S. Im, J. Park, H.-G. Jeon, and I. S. Kweon. High-quality depth from uncalibrated small motion clip. In *CVPR*, 2016. 1

[22] C. Hane, L. Ladicky, and M. Pollefeys. Direction matters: Depth estimation with a surface normal classifier. In *CVPR*, 2015. 2

[23] F. E. Harrell Jr. *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*. Springer, 2015. 3

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 5

[25] R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. 1999. 3

[26] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 75(1):151–172, 2007. 1, 2

[27] 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

[28] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE TPAMI*, 36(11):2144–2158, 2014. 1, 2, 6, 7

[29] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*, 2017. 2

[30] S. Kim, K. Park, K. Sohn, and S. Lin. Unified depth prediction and intrinsic image decomposition from a single image via joint convolutional neural fields. In *ECCV*, 2016. 1, 2, 7

[31] N. Kong and M. J. Black. Intrinsic depth: Improving depth transfer with intrinsic images. In *ICCV*, 2015. 1

[32] J. Konrad, M. Wang, P. Ishwar, C. Wu, and D. Mukherjee. Learning-based, automatic 2d-to-3d image and video conversion. *IEEE TIP*, 22(9):3485–3496, 2013. 2

[33] Y. Kuznietsov, J. Stückler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. *CVPR*, 2017. 1, 2, 3, 7

[34] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *CVPR*, 2014. 1, 2, 7

[35] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016. 1, 2, 3, 7

[36] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *CVPR*, 2015. 7

[37] J. Li, R. Klein, and A. Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *ICCV*, 2017. 2, 7

[38] X. Li, H. Qin, Y. Wang, Y. Zhang, and Q. Dai. Dept: depth estimation by parameter transfer for single still images. In *ACCV*, 2014. 2

[39] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *CVPR*, 2010. 7

[40] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE TPAMI*, 38(10):2024–2039, 2016. 1, 2, 3, 5, 6, 7

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

[42] T. Narihira, M. Maire, and S. X. Yu. Learning lightness from human judgement on relative reflectance. In *CVPR*, 2015. 2

[43] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 1, 2, 5, 6

[44] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua. Ordinal regression with multiple output cnn for age estimation. In *CVPR*, 2016. 3

[45] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001. 2

[46] A. Rajagopalan, S. Chaudhuri, and U. Mudenagudi. Depth estimation and image restoration using defocused stereo pairs. *IEEE TPAMI*, 26(11):1521–1525, 2004. 1

[47] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun. Dense monocular depth estimation in complex dynamic scenes. In *CVPR*, 2016. 2

[48] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *CVPR*, 2016. 1, 2, 7

[49] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 5

[50] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *NIPS*, 2006. 1, 2, 5, 6

[51] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE TPAMI*, 31(5):824–840, 2009. 1, 2, 5, 6, 7

[52] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002. 2

[53] A. Shashua and A. Levin. Ranking with large margin principle: Two approaches. In *NIPS*, 2003. 3

[54] E. Shelhamer, J. T. Barron, and T. Darrell. Scene intrinsics and depth from a single image. In *ICCV Workshop*, 2015. 2

[55] J. Shi, X. Tao, L. Xu, and J. Jia. Break ames room illusion: depth from general single images. *ACM TOG*, 34(6):225, 2015. 2

[56] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2, 5

[57] 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, 4, 7

[58] X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *CVPR*, 2015. 2

[59] J. Xie, R. Girshick, and A. Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *ECCV*, 2016. 2, 3, 7

[60] 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. 7

[61] X. You, Q. Li, D. Tao, W. Ou, and M. Gong. Local metric learning for exemplar-based object detection. *IEEE TCSVT*, 24(8):1265–1276, 2014. 2

[62] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2, 3

[63] Z. Zhang, A. G. Schwing, S. Fidler, and R. Urtasun. Monocular object instance segmentation and depth ordering with cnns. In *ICCV*, 2015. 2

[64] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017. 2, 3

[65] W. Zhuo, M. Salzmann, X. He, and M. Liu. Indoor scene structure analysis for single image depth estimation. In *CVPR*, 2015. 2

[66] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman. Learning ordinal relationships for mid-level vision. In *ICCV*, 2015. 3